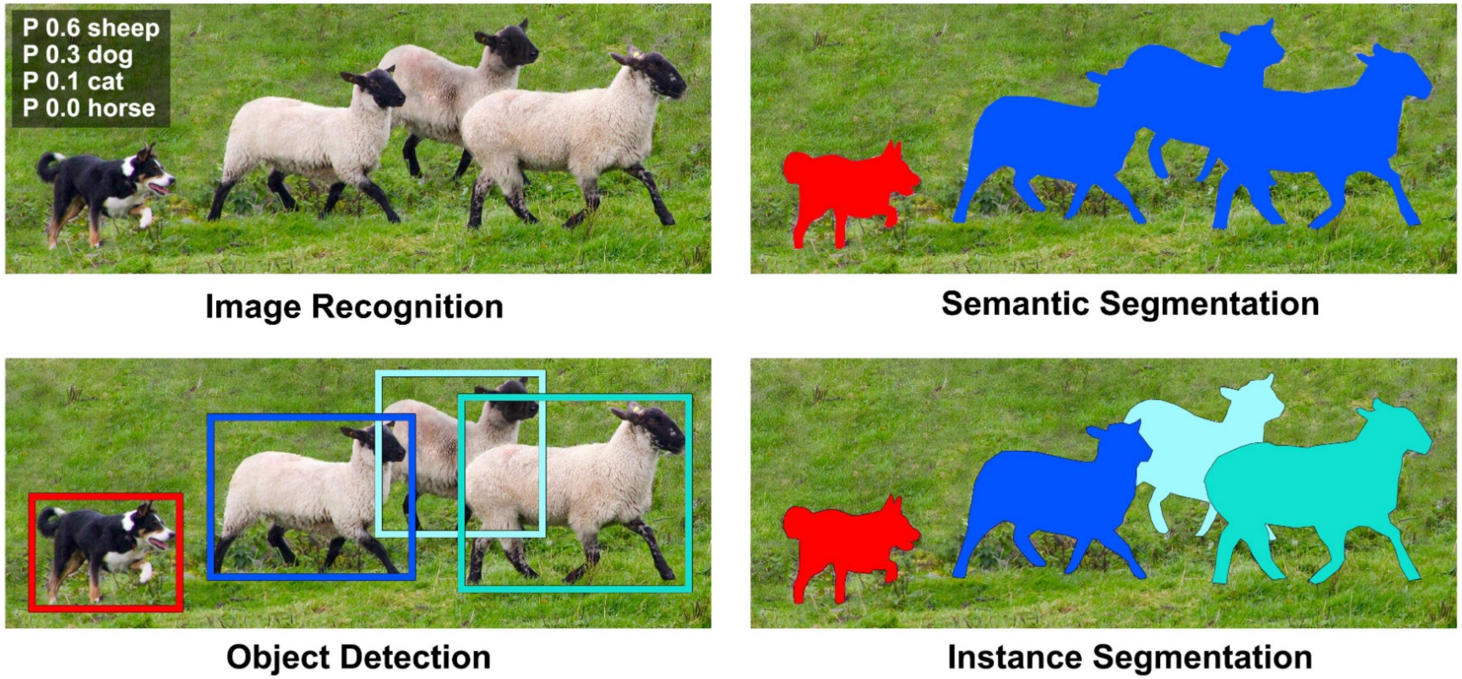
Weakly Supervised Nuclei Segmentation via Instance Learning

Weizhen Liu

Xuan Gao

Qiyuan Ma

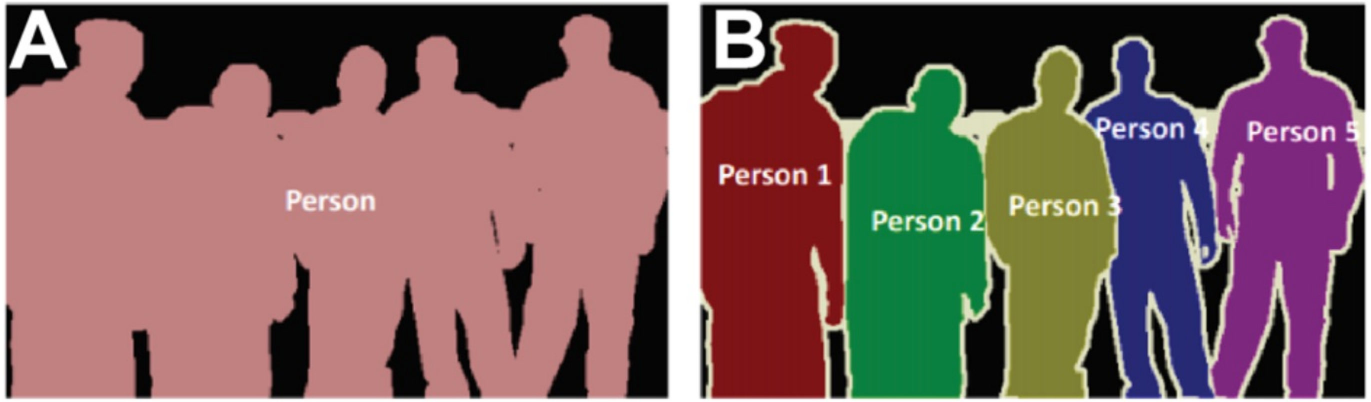
Instance Segmentation



Instance segmentation

- Characteristic of semantic segmentation: pixel-level classification
- Characteristic of object detection: locating difference instances, even if they are of the same class

Instance Segmentation

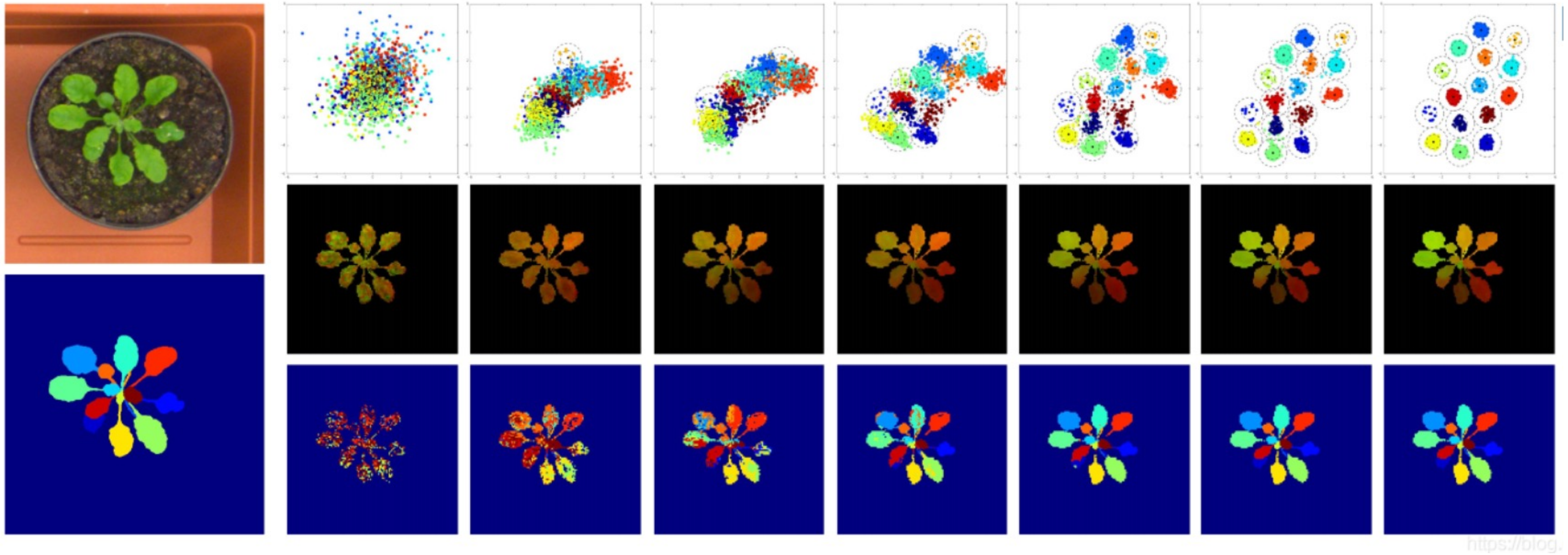


Semantic Segmentation Instance Segmentation

- **<u>Top-down detection-based</u>**

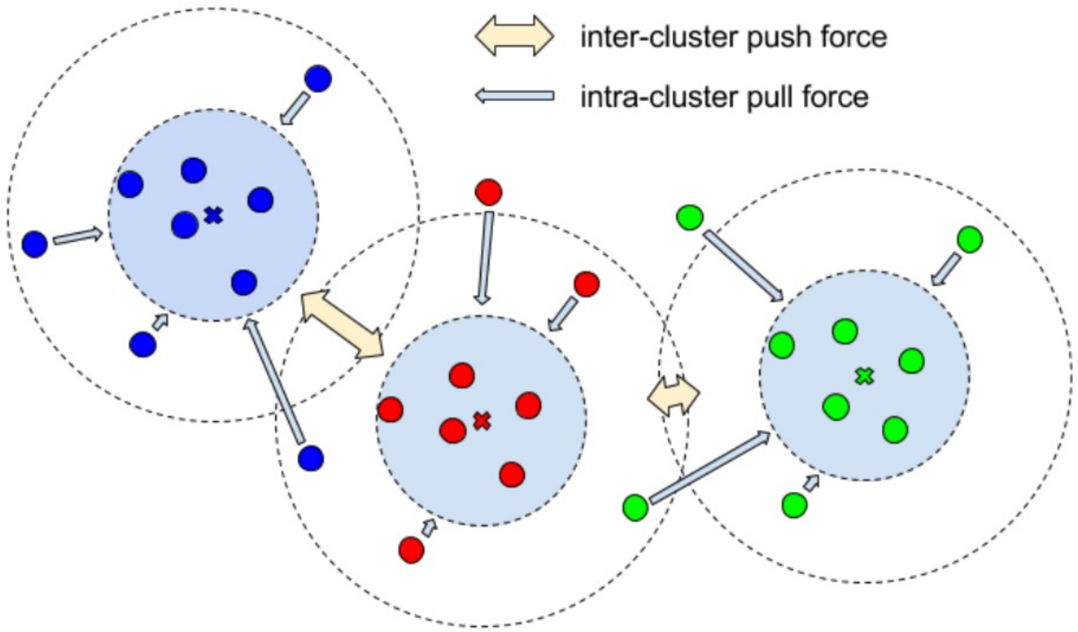
- **<u>Bottom up:</u>**

1. Perform semantic segmentation at the pixel level
2. Distinguish different instances through clustering and other methods.



- (1) Semantic Segmentation
- (2) Pixel Embedding: a discriminative loss function is used to train the network, where the optimization goal is to project each pixel of the image into an n-dimensional feature space
- (3) Post-processing: Distinguish different instances through clustering and other methods. mean-shift

Loss Function



- (1) variance term: an intra-cluster pull-force that draws embeddings towards the mean embedding, i.e. the cluster center.
- (2) distance term: an inter-cluster push-force that pushes clusters away from each other, increasing the distance between the cluster centers.
- (3) regularization term: a small pull-force that draws all clusters towards the origin, to keep the activations bounded. The center points should not be too far away from the origin.

Bert De Brabandere, Davy Neven, and Luc Van Gool, "Semantic instance segmentation with a discriminative loss function," arXiv preprint arXiv:1708.02551, 2017.

Discriminative Loss

$$L_{var} = \frac{1}{C} \sum_{c=1}^{C} \frac{1}{N_c} \sum_{i=1}^{N_c} \left[\| \mu_c - x_i \| - \delta_{\mathrm{v}} \right]_+^2 \qquad (1)$$

$$L_{dist} = \frac{1}{C(C-1)} \sum_{\substack{c_A=1 \\ c_A \neq c_B}}^{C} \sum_{c_B=1}^{C} \left[2\delta_{\mathrm{d}} - \| \mu_{c_A} - \mu_{c_B} \| \right]_+^2 \qquad (2)$$

$$L_{reg} = \frac{1}{C} \sum_{c=1}^{C} \| \mu_c \| \qquad (3)$$

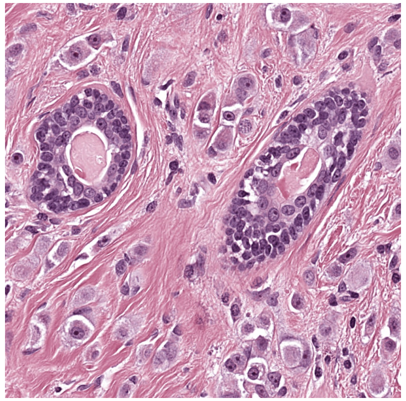
$$L = \alpha \cdot L_{var} + \beta \cdot L_{dist} + \gamma \cdot L_{reg} \qquad (4)$$

Bert De Brabandere, Davy Neven, and Luc Van Gool, "Semantic instance segmentation with a discriminative loss function," arXiv preprint arXiv:1708.02551, 2017.

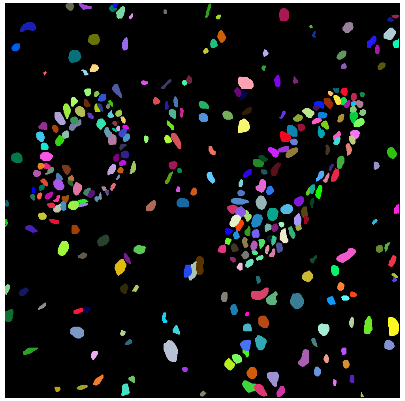
Dataset

- MultiOrgan
  - 30 images of size 1000 × 1000
  - 16 training, 14 testing
  - Multiple organs: breast, liver, kidney…
  - https://github.com/weizhenFrank/WeakNucleiSeg/blob/main/data/MO/train_val_test.json
- TNBC
  - 50 images of size 512 × 512
  - 37 training, 13 testing
  - Breast
  - https://github.com/weizhenFrank/WeakNucleiSeg/blob/main/data/TNBC/train_val_test.json

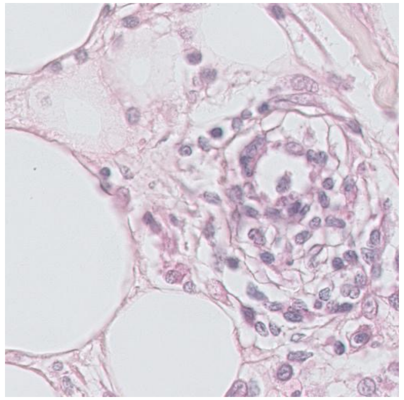
Image



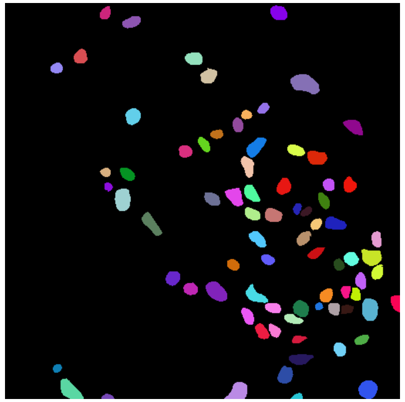
Mask



Image



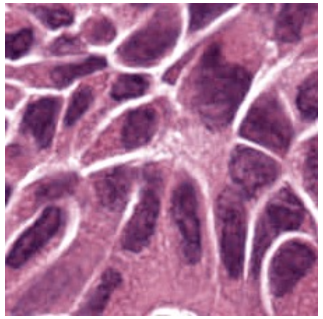
Mask



Problem Definition

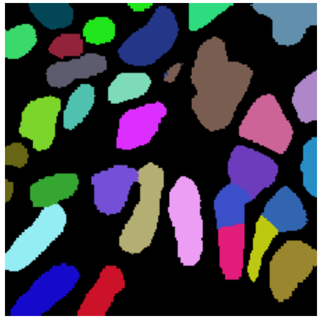
- Nuclei Segmentation

Input Image



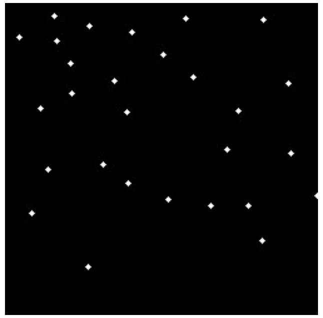
Segment

Output Mask



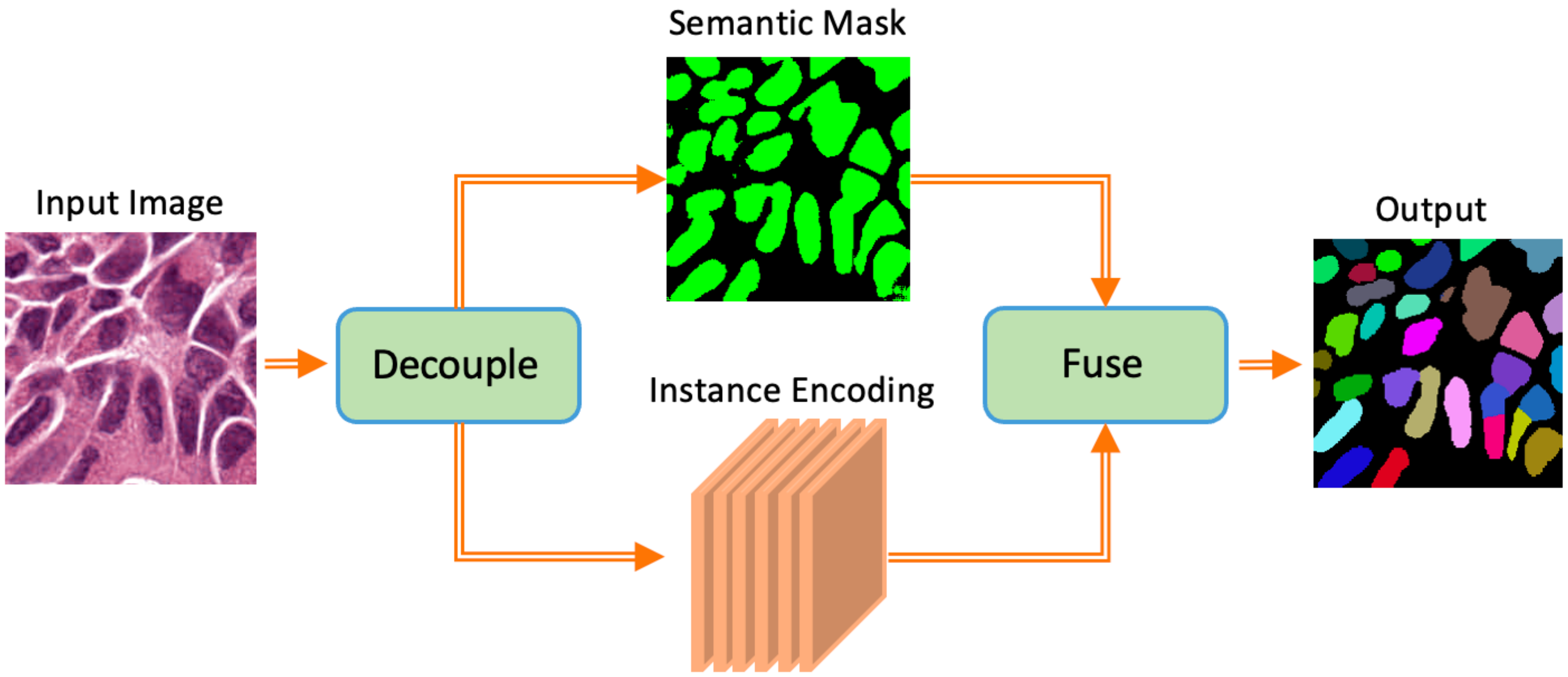
- Weak Label – point

Point Label



Main Idea

Decouple and then fuse semantic and instance segmentation



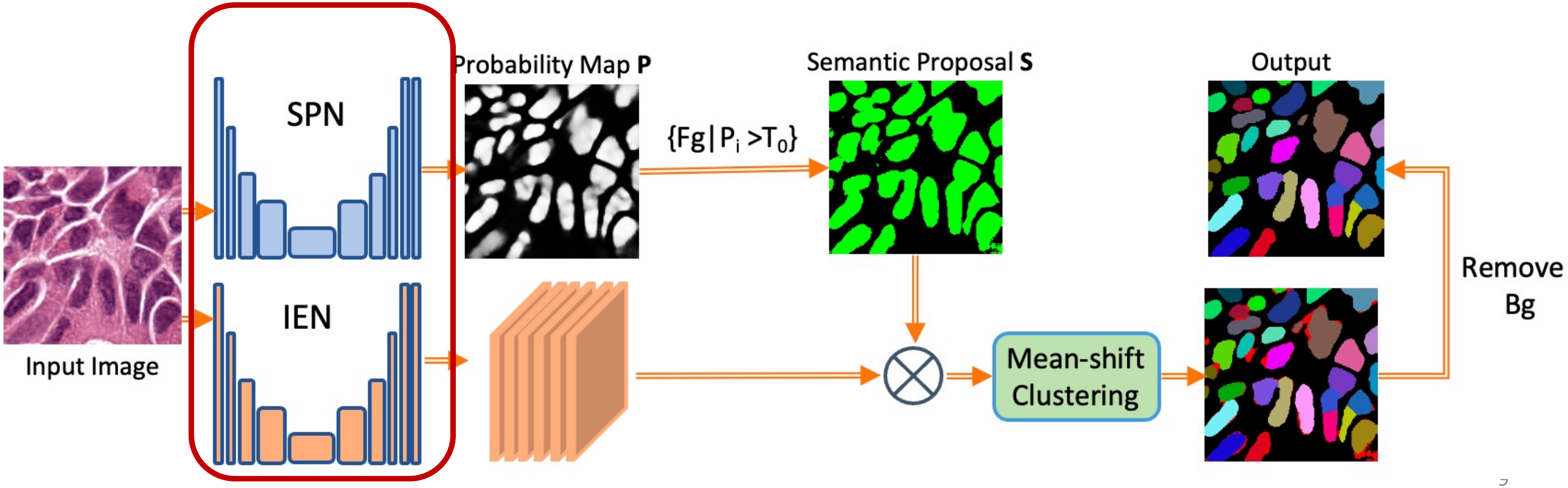
Model Design

Our model consists of two branches:

- SPN – Semantic Proposal Network
- IEN – Instance Encoding Network

Given an input image:

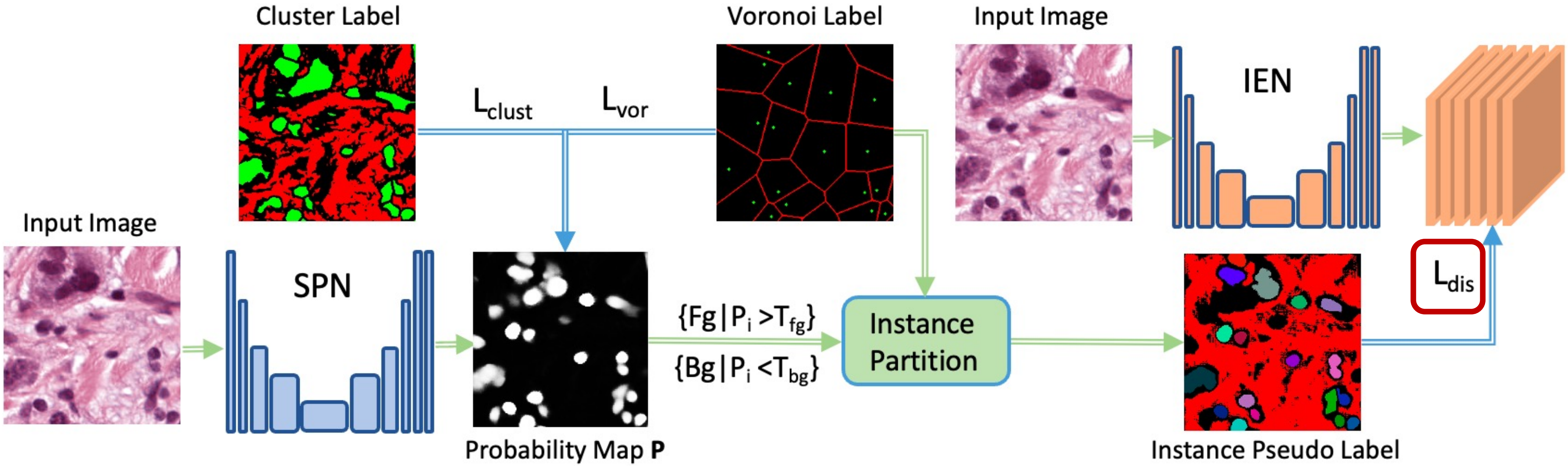
- SPN – generates semantic proposal
- IEN – generates instance encoding



Model Training

- Train SPN with cluster label and voronoi label
  - Lclust and Lvor are cross-entropy losses on partially labeled pixels

- Train IEN with instance pseudo label
  - L_{dis} is a discriminative loss for instance-aware representation learning



Results

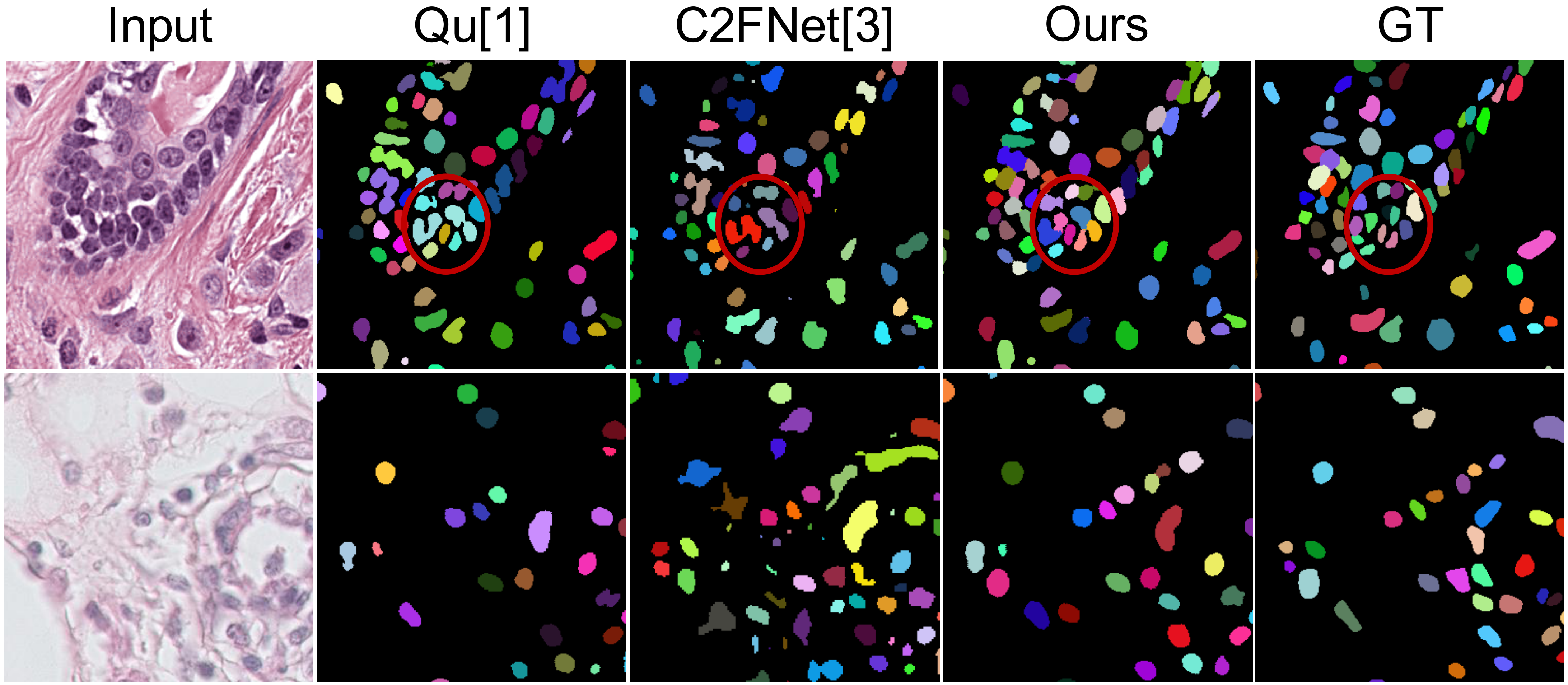
- Ten-fold cross-validation on MultiOrgan and TNBC

Method	IoU	F1	Dice	AJI
MultiOrgan				
[3]	0.6136±0.04	-	-	-
[1]	0.5789±0.06	0.7320±0.05	0.7021±0.04	0.4964±0.06
[4]	0.6239±0.03	0.7638±0.02	0.7132±0.02	0.4927±0.04
Ours	**0.6494 ± 0.02**	**0.7863 ± 0.02**	**0.7394 ± 0.02**	**0.5430 ± 0.04**
TNBC				
[3]	0.6038±0.03	-	-	-
[1]	0.5420±0.04	0.7008±0.04	0.6931±0.04	0.5181±0.05
[4]	**0.6393 ± 0.03**	0.7510±0.04	0.7413±0.03	0.5509±0.04
Ours	0.6153 ± 0.03	**0.7600 ± 0.02**	**0.7492 ± 0.02**	**0.5854 ± 0.03**

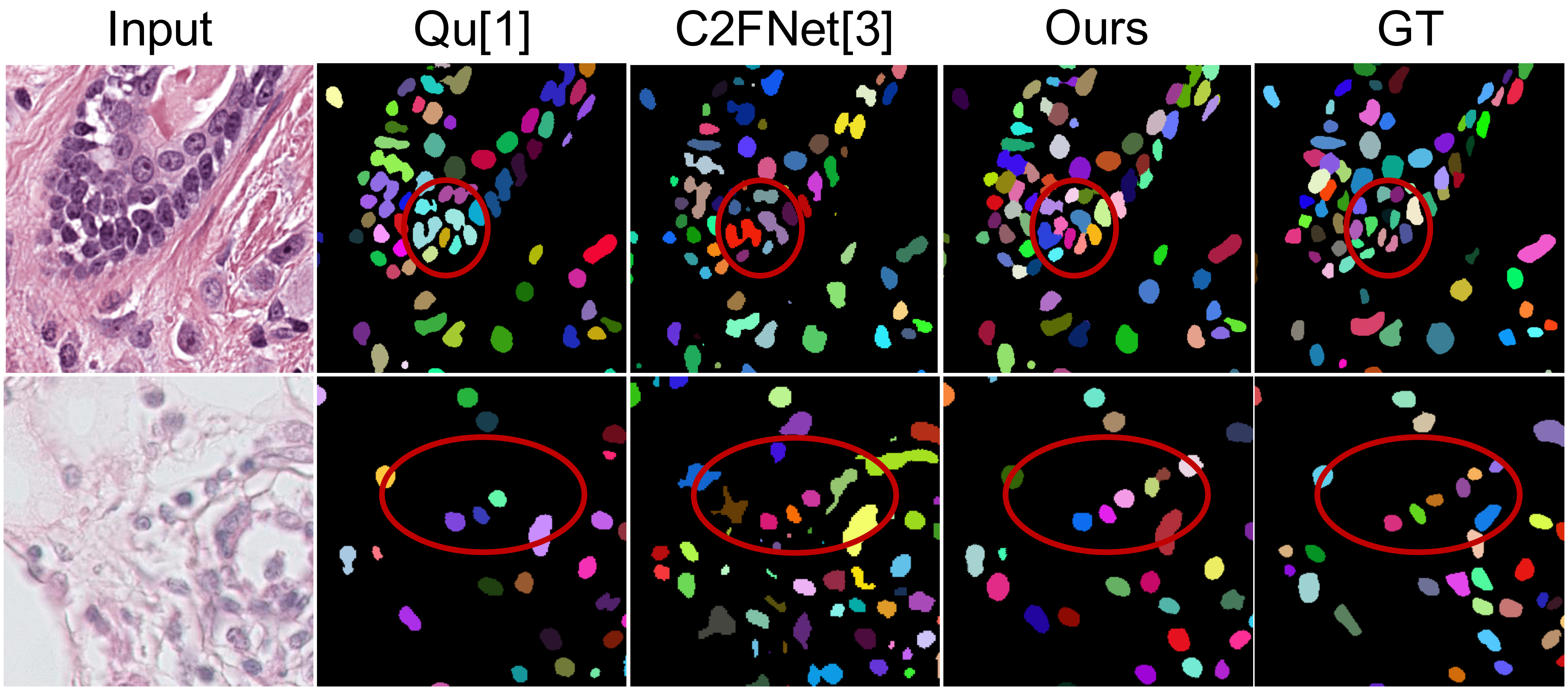
+5.03% in AJI

+3.45% in AJI

Visualization



Visualization



Partial Point annotation?

- Question:
  - Can we extend our method to partial points annotation?

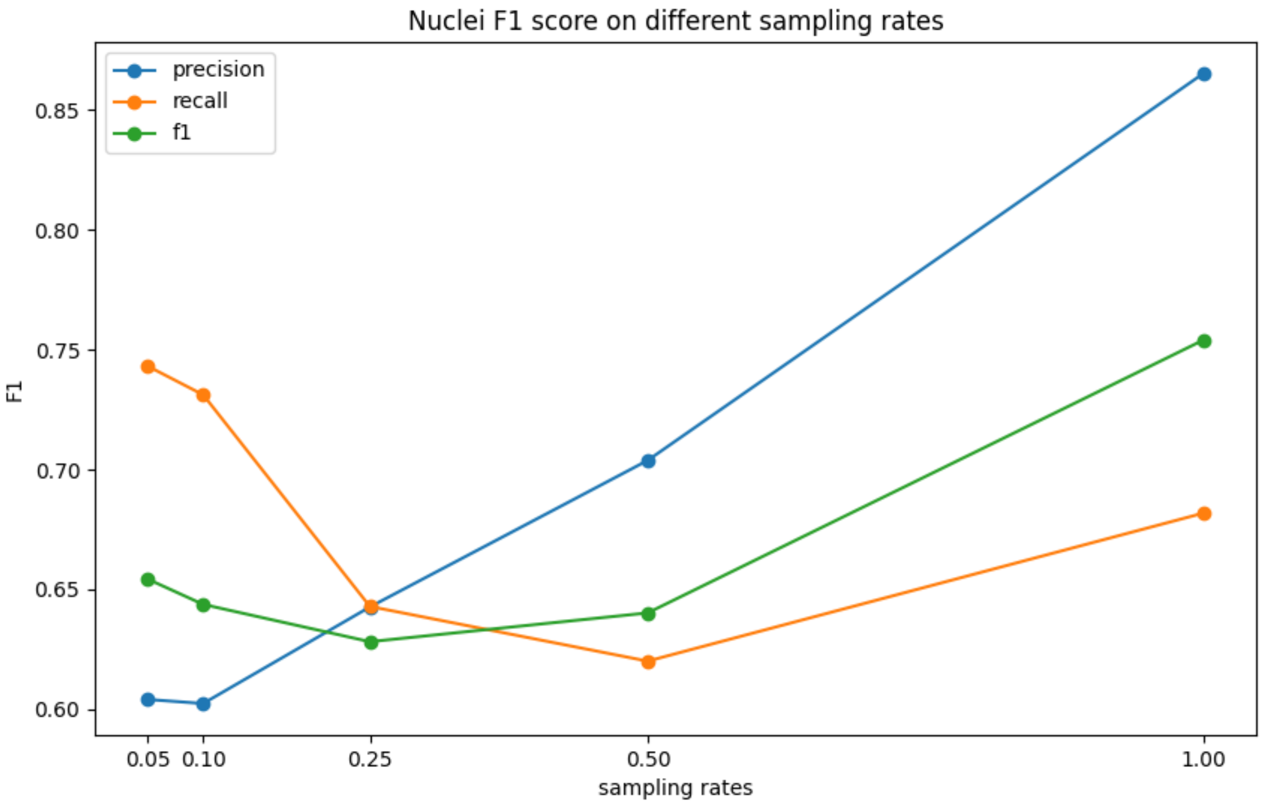
- Why partial points annotation?
  - Even annotate each nuclei with a point is not easy, hundreds of points.
  - We can’t always fully annotate all the nuclei with points, human error

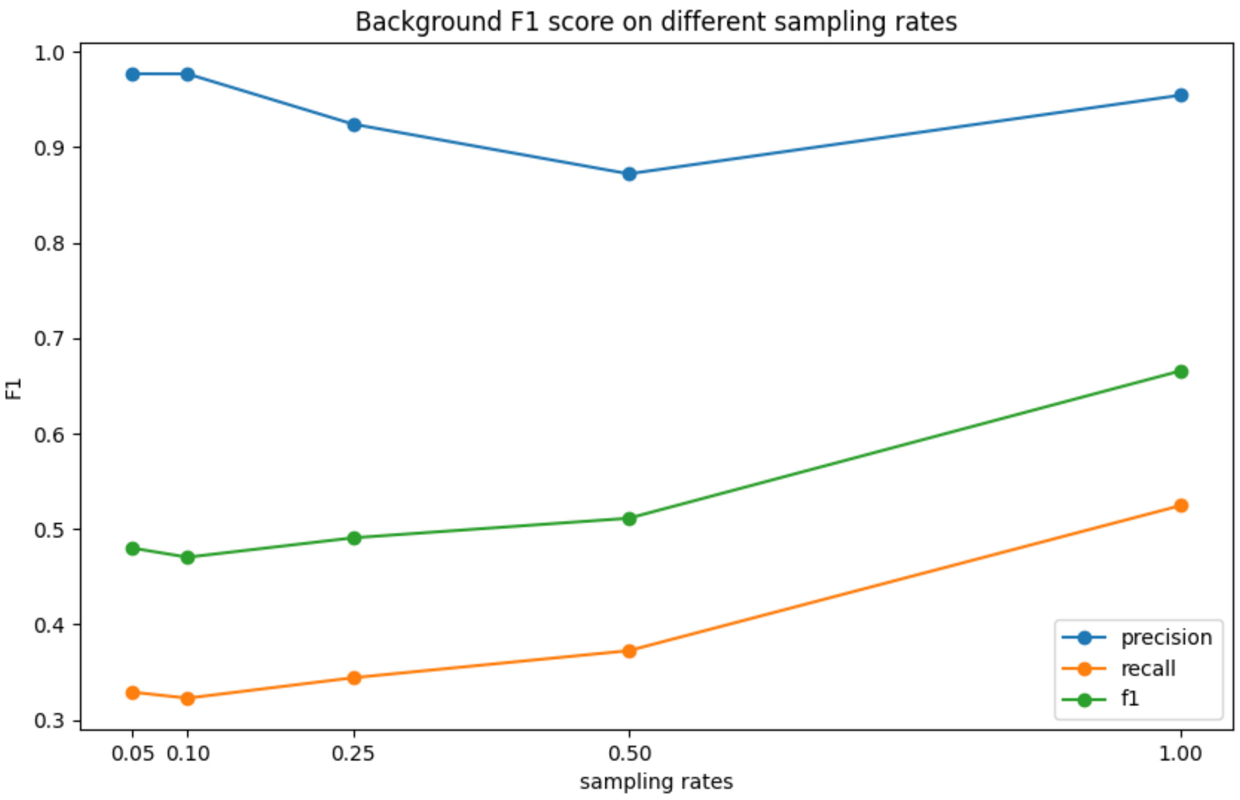
Analysis on partial points

- Pseudo label quality
  - Label from K-means clustering
- Semantic learning results
  - IOU, F1 metrics

Pseudo label quality

- Pseudo label quality
  - Label from K-means clustering

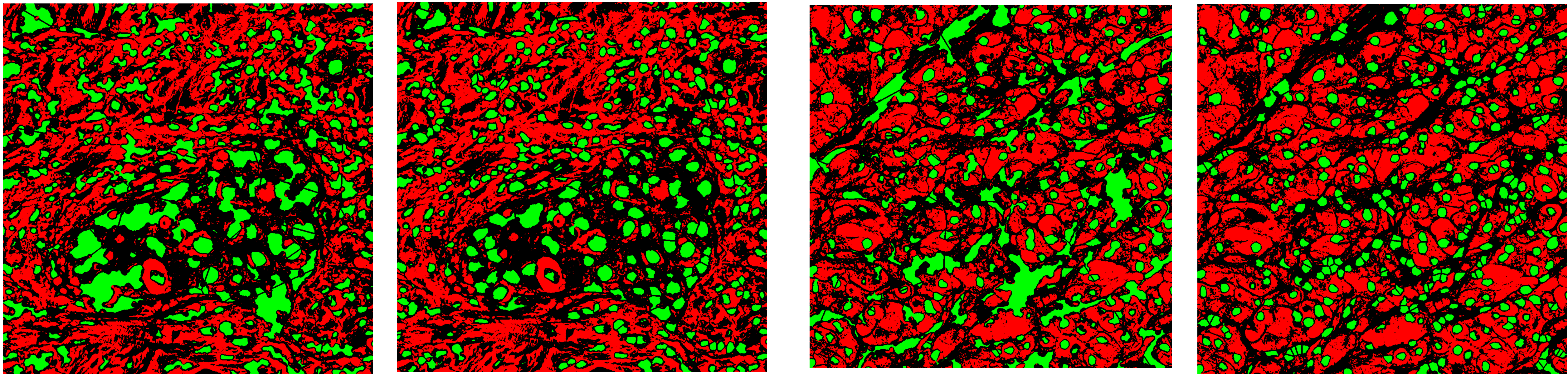




Pseudo label quality

- Pseudo label quality
  - Label from K-means clustering

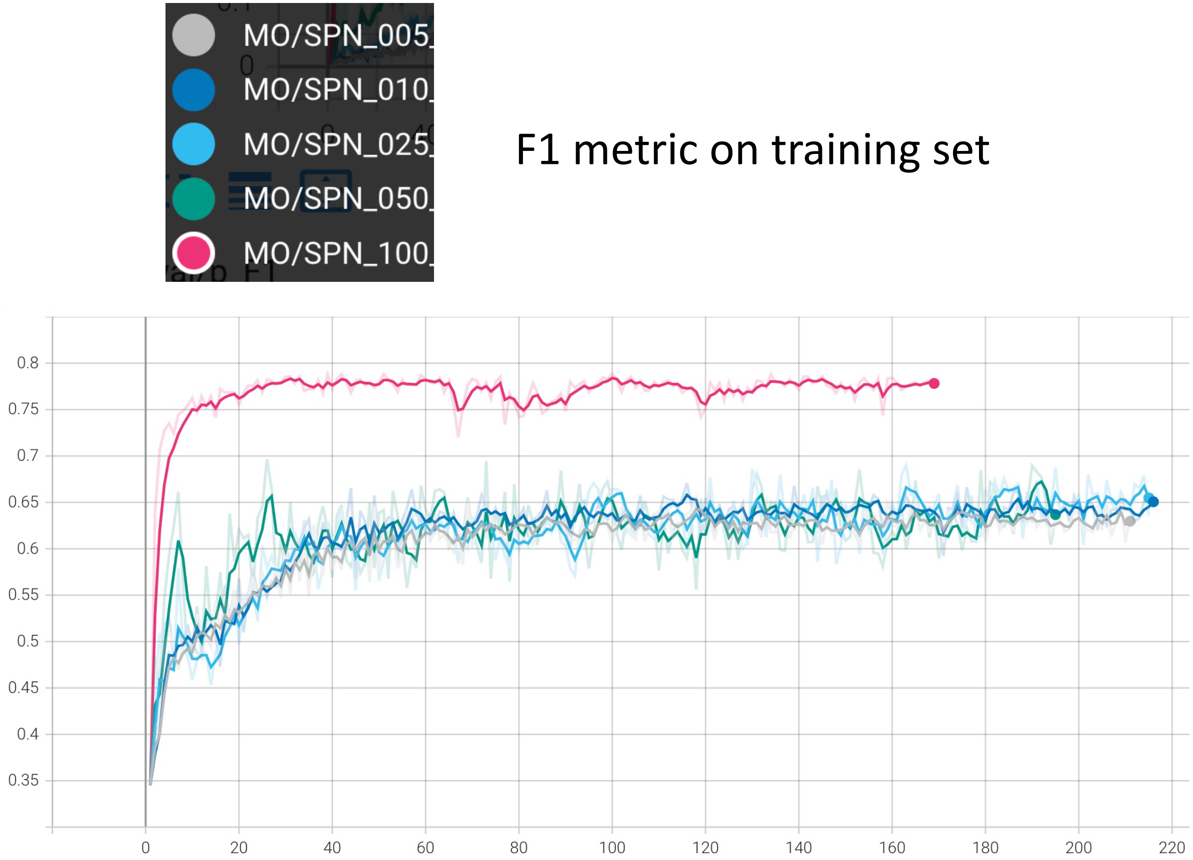
Sampling rate = 0.05 | Sampling rate = 1 | Sampling rate = 0.05 | Sampling rate = 1



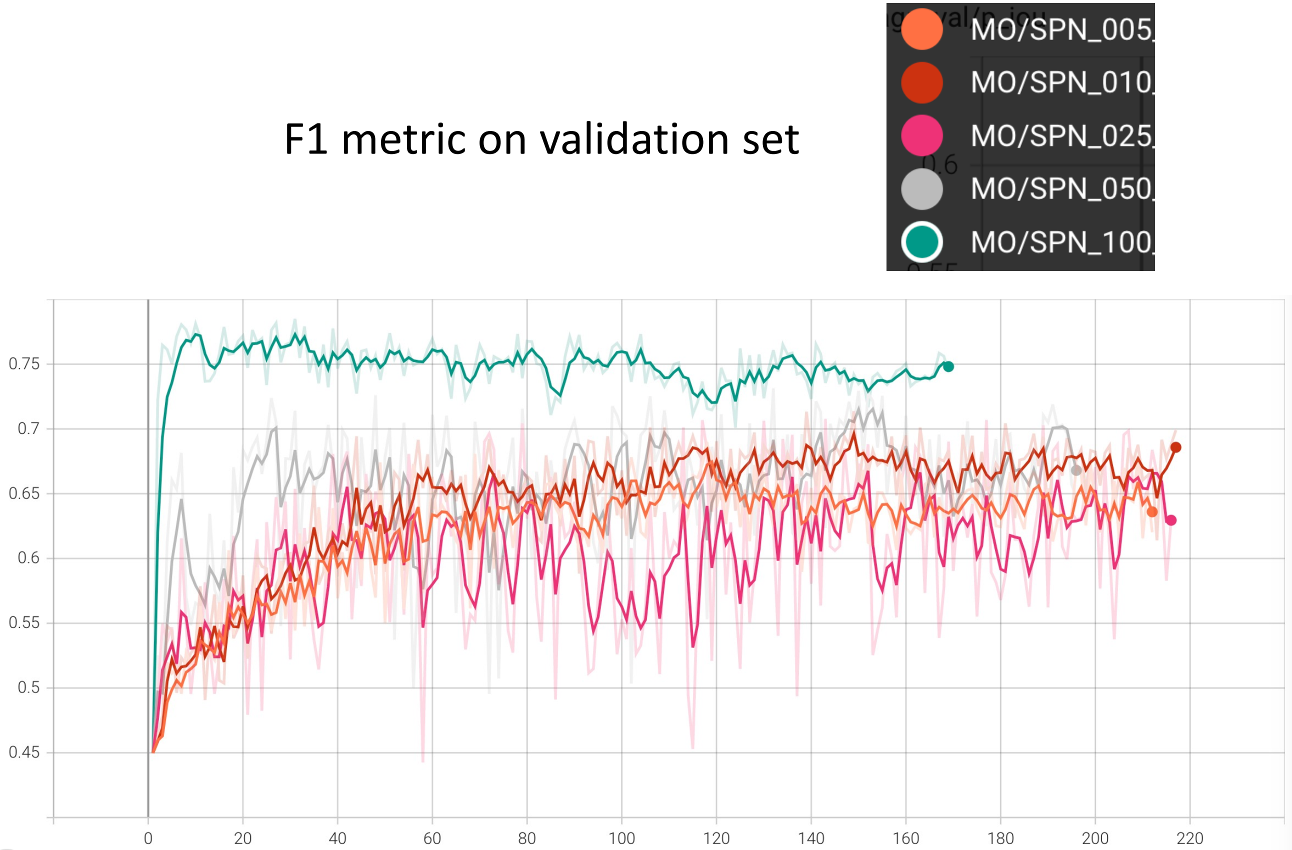
Pseudo label quality

- Semantic learning results

F1 metric on training set

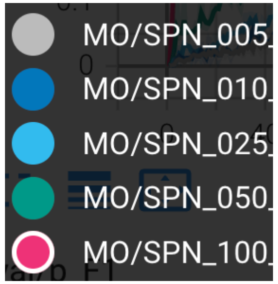


F1 metric on validation set

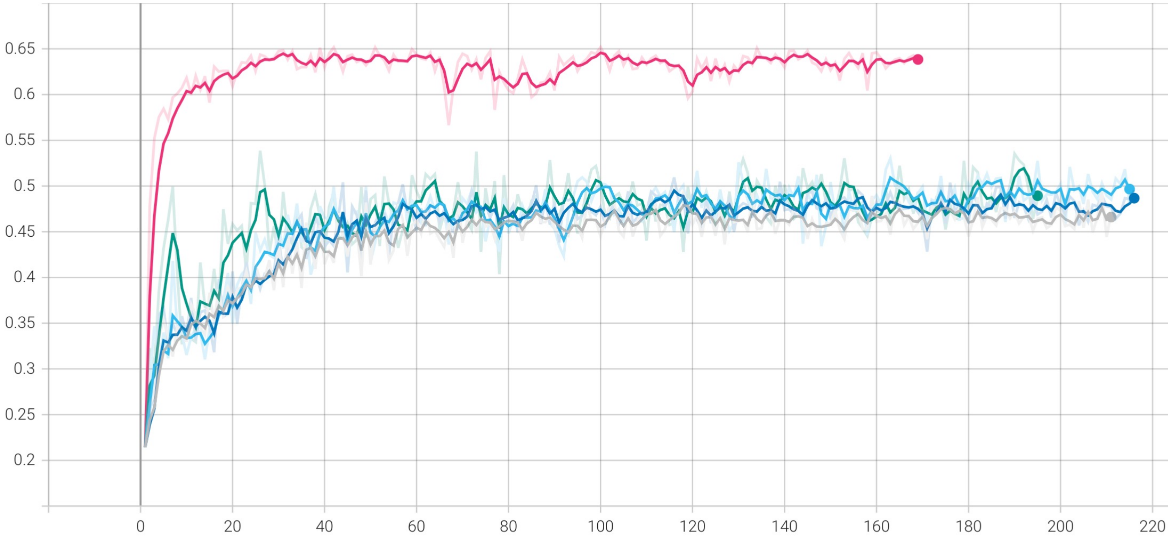


Pseudo label quality

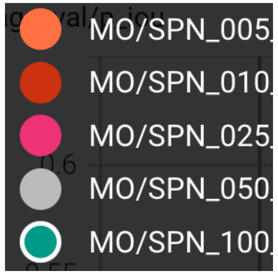
- Semantic learning results

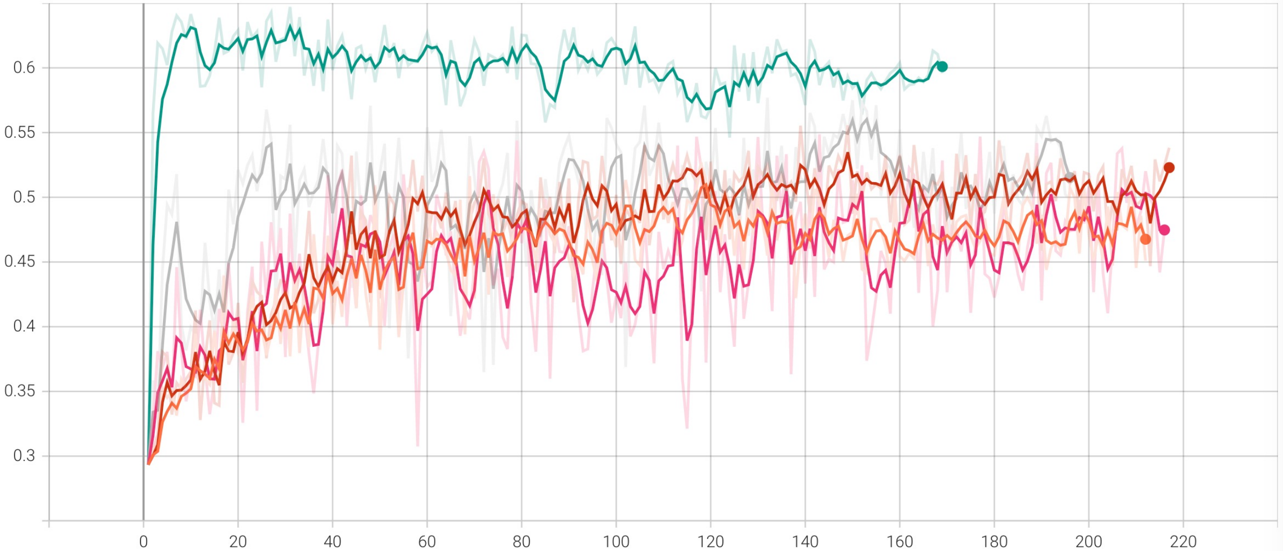


IOU metric on training set



IOU metric on validation set





Where's the problem from

- Low quality of K means clustering label

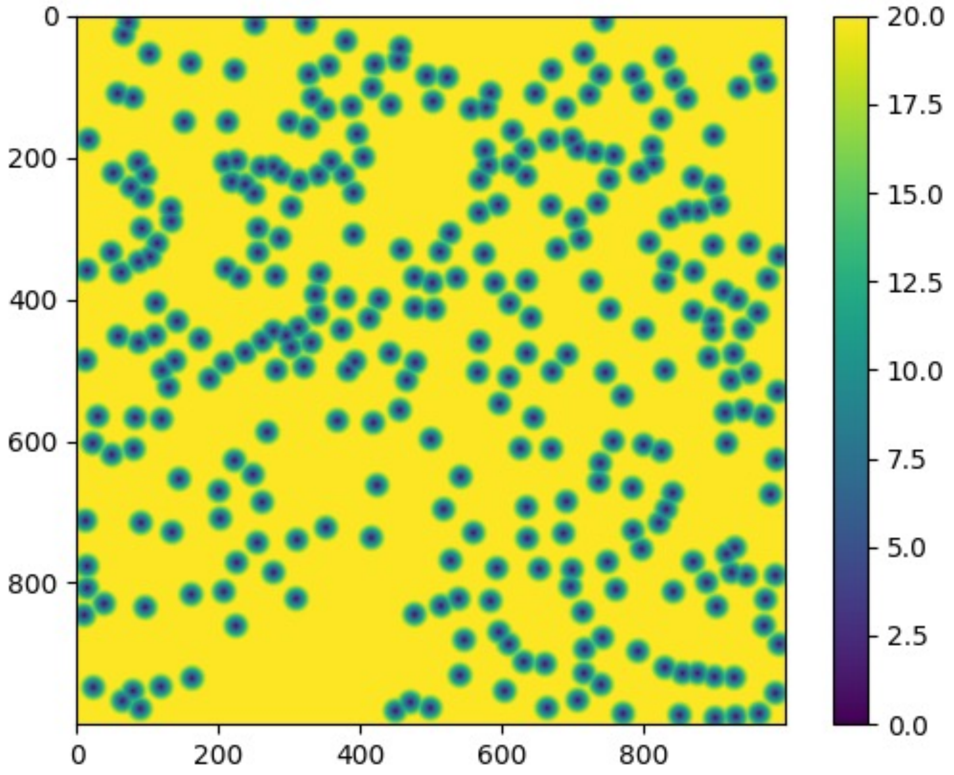
Given a set of observations ($\mathbf{x}_1$, $\mathbf{x}_2$, ..., $\mathbf{x}_n$), where each observation is a *d*-dimensional real vector, *k*-means clustering aims to partition the *n* observations into *k* ($\leq n$) sets $\mathbf{S} = \{S_1, S_2, ..., S_k\}$ so as to minimize the within-cluster sum of squares (WCSS)

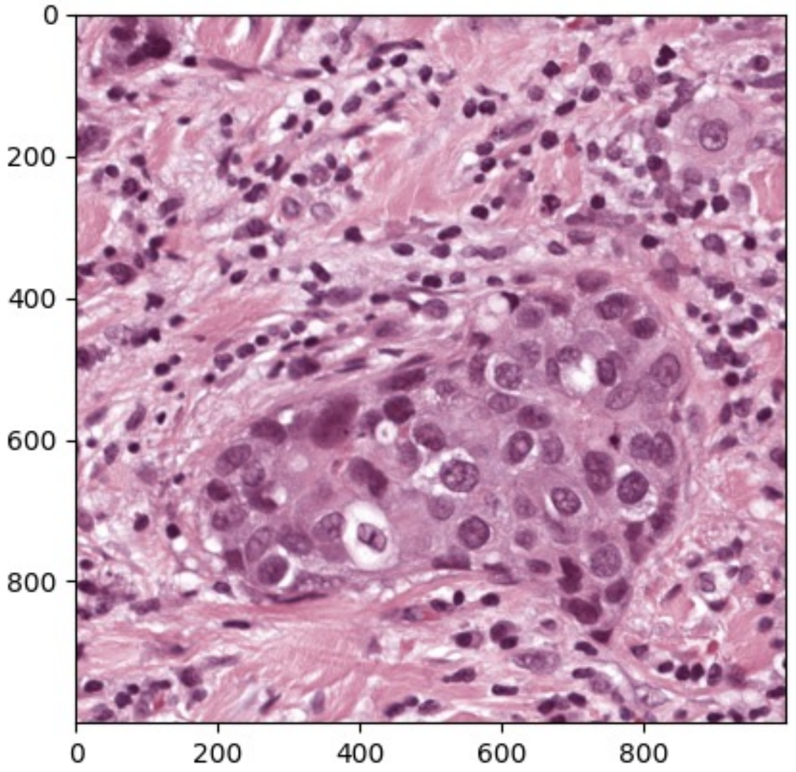
$$\underset{\mathbf{S}}{\arg\min} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \left\| \mathbf{x} - \boldsymbol{\mu}_i \right\|^2 = \underset{\mathbf{S}}{\arg\min} \sum_{i=1}^{k} |S_i| \operatorname{Var} S_i$$

Where's the problem from

- Our x feature vector comes
  - Distance map to the point
  - RGB color image

$$X = [Dist|\boldsymbol{\alpha} * RGB/\beta]$$





We need to tune $\boldsymbol{\alpha}$ and β!

Grid search for best alpha, and beta for different nuclei sampling rate

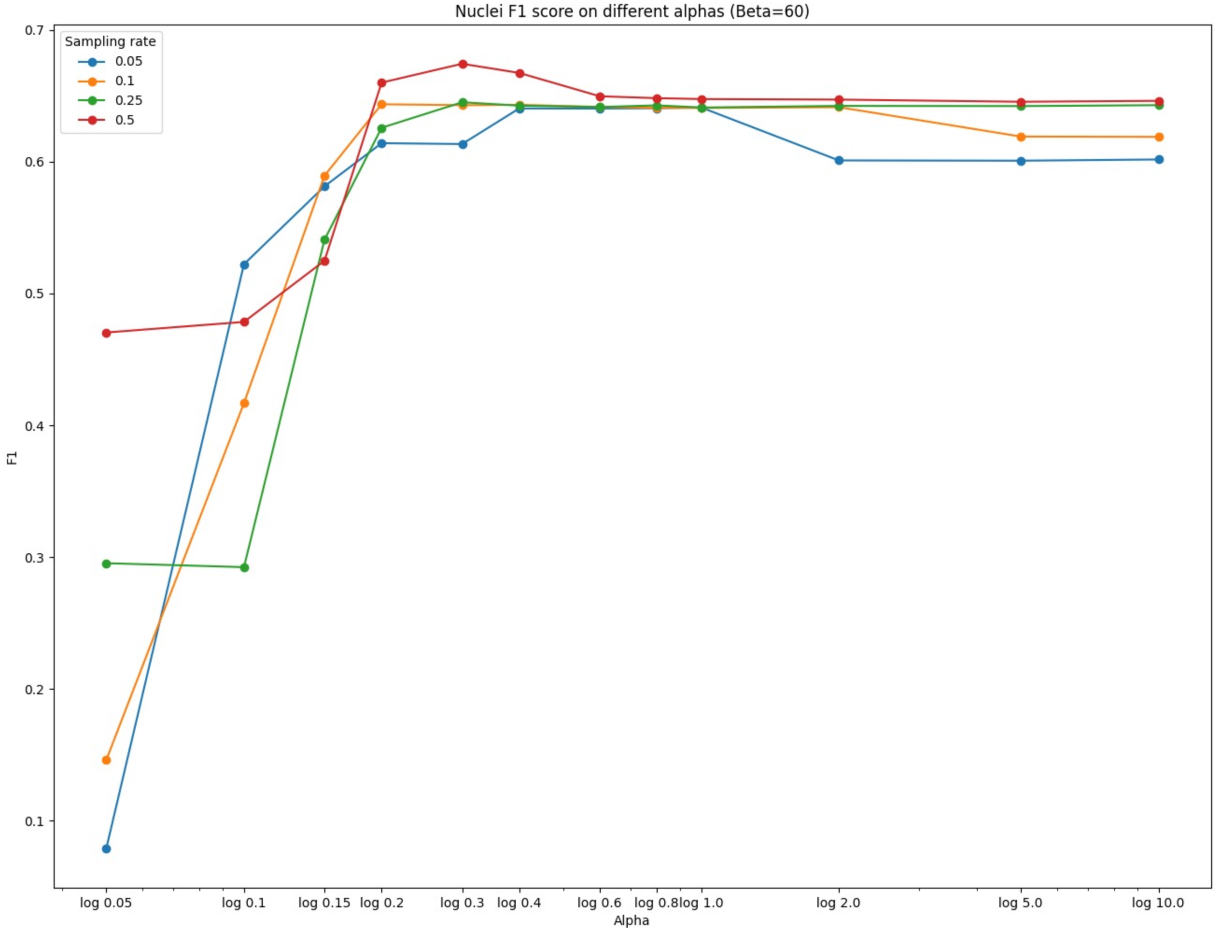
$$X = [Dist|\boldsymbol{\alpha} * RGB/\beta]$$

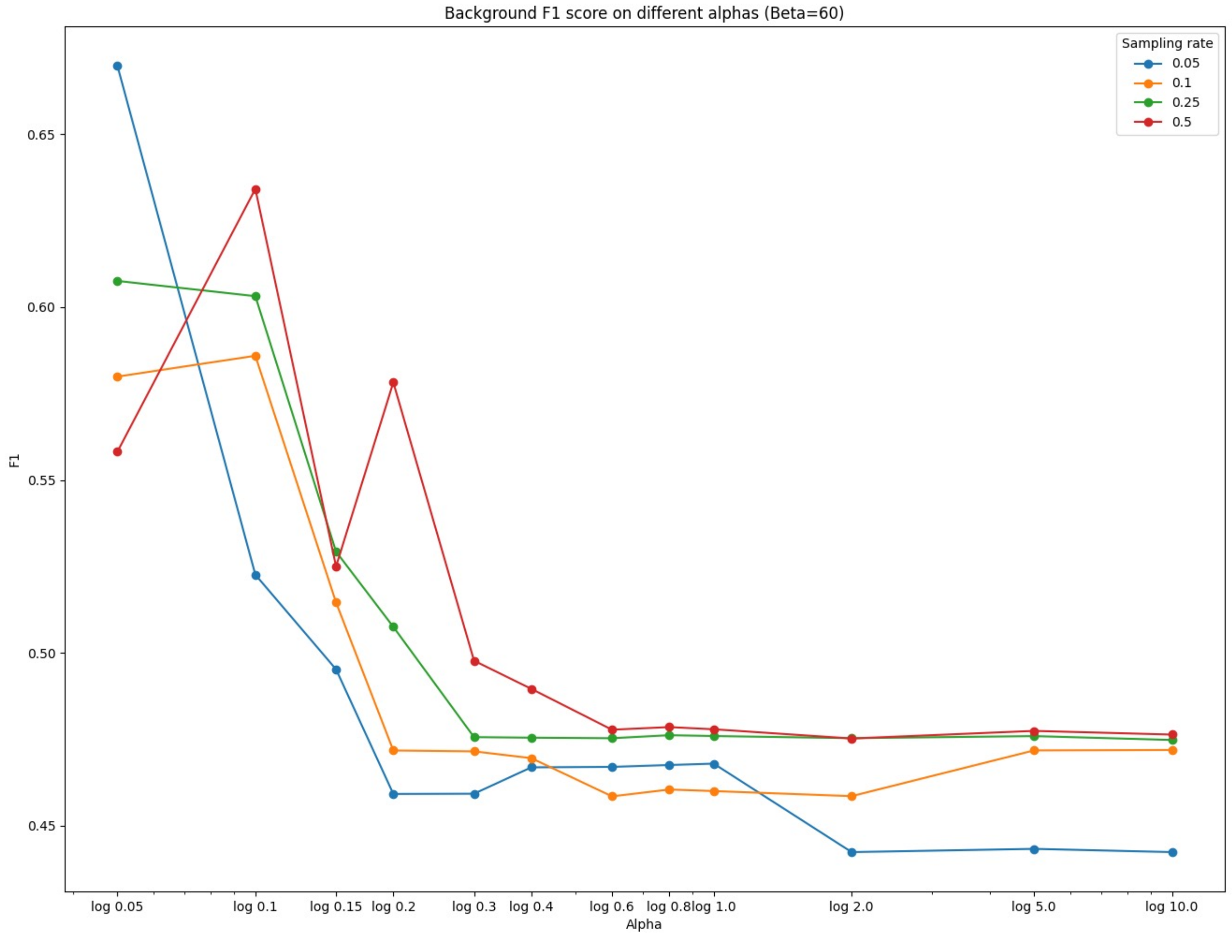
Alpha: The balance coefficient
[0.05, 0.1, 0.15, **0.2,** 0.3, 0.4, 0.6, 0.8, 1, 2, 5, 10]

Beta: Kernel size to calculate the the image statistics
[10, 20, 40, **60**, 80, 100, 120]

We need to tune $\boldsymbol{\alpha}$ and β!

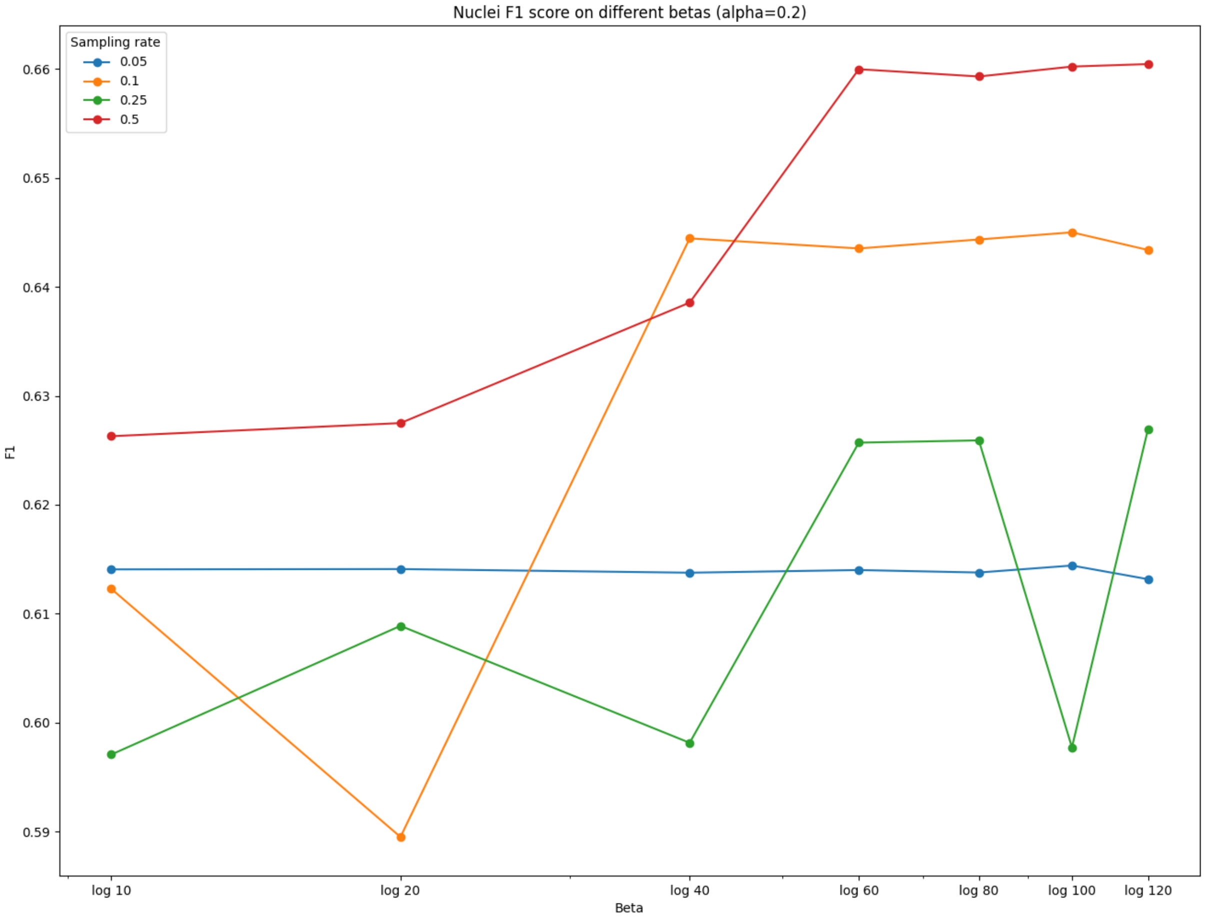
$$X = [Dist|\boldsymbol{\alpha} * RGB/\beta]$$

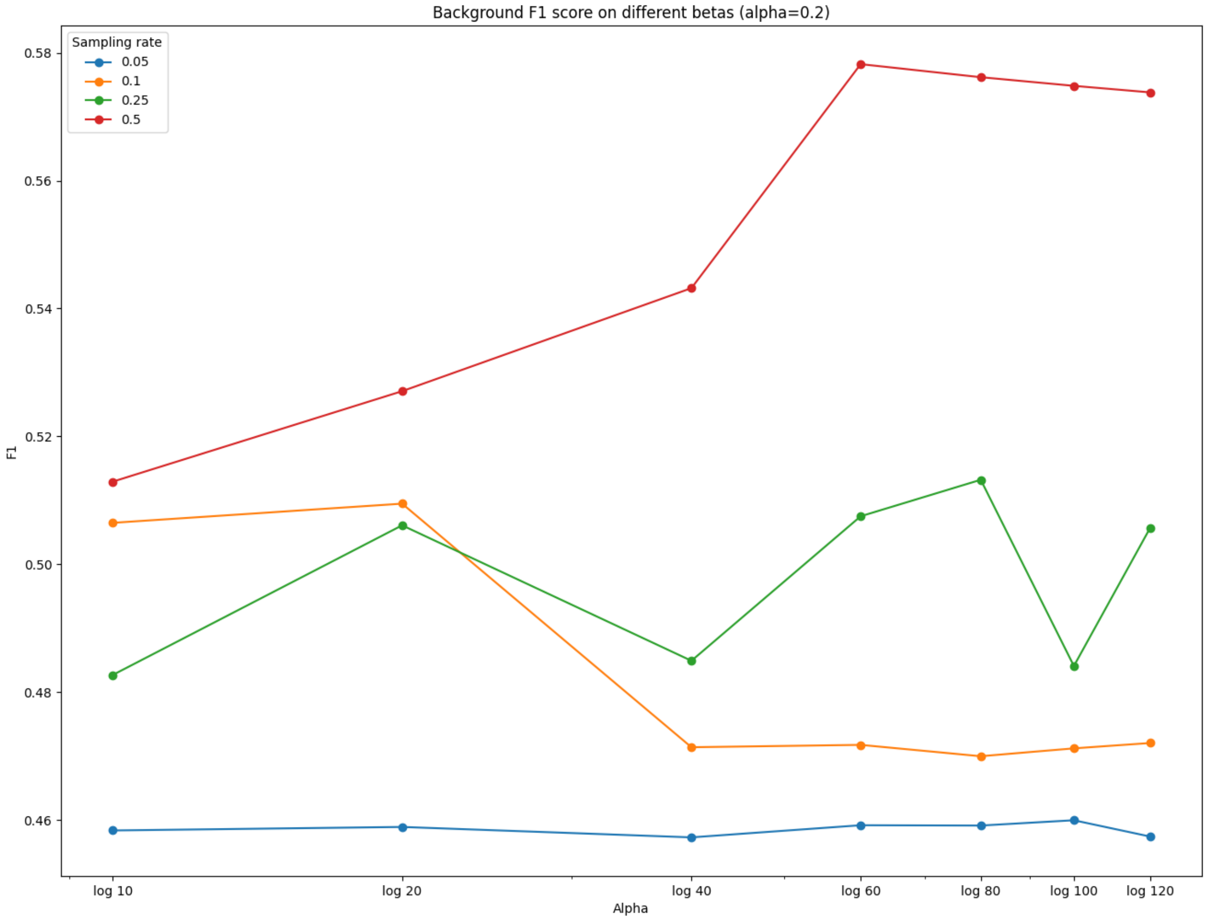




We need to tune $\boldsymbol{\alpha}$ and β!

$$X = [Dist|\boldsymbol{\alpha} * RGB/\beta]$$





Conclusion

- Our method achieve the best results on full points annotation
    - Significant improvement in instance metrics on two benchmarks

- Extension to partial points annotation
    - Good news: the balance method quite stable for different sampling
    - Bad news: even under best tuned hyper-parameter, the pseudo label quality still bad

Raw ideas: limited by time

- Need to design a de novo framework to handle partial points annotation problem
    - Qu et al., IEEE Transactions on Medical Imaging, 2020: Predict the absent points
    - Human in the loop, interactive instance segmentation

Q & A