# Medical Vision Seminar

Luyue Shi

Aug 11, 2021

(MICCAI2020) Meta Corrupted Pixels Mining for Medical Image Segmentation

(MICCAI2021) Distilling effective supervision for robust medical image segmentation with noisy labels

(MICCAI2020)

# Meta Corrupted Pixels Mining for Medical Image Segmentation

Jixin Wang[1], Sanping Zhou[1], Chaowei Fang[2], Le Wang[1], and Jinjun Wang[1(✉)]

[1] Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, Shaanxi, People's Republic of China
jinjun@mail.xjtu.edu.cn

[2] School of Artificial Intelligence, Xidian University, Xi'an, Shaanxi, People's Republic of China

## Introduction

Problem

In medical image segmentation, it is very laborious and expensive to acquire precise pixel-level annotations. This paper aims at training deep segmentation models on datasets with probably corrupted annotations

Contribution:

a. A novel meta learning framework to mine pixels with corrupted labels during the process of training a segmentation network.

b. Based on the fully convolutional structure, we build up a meta mask network which can automatically estimate pixel-wise importance factors for mitigating the influence of corrupted labels.

c. Extensive experiments on both LIDC-IDRI and LiTS datasets indicate that our method achieves the state-of-the-art performance in medical image segmentation with incorrect labels.

# Introduction

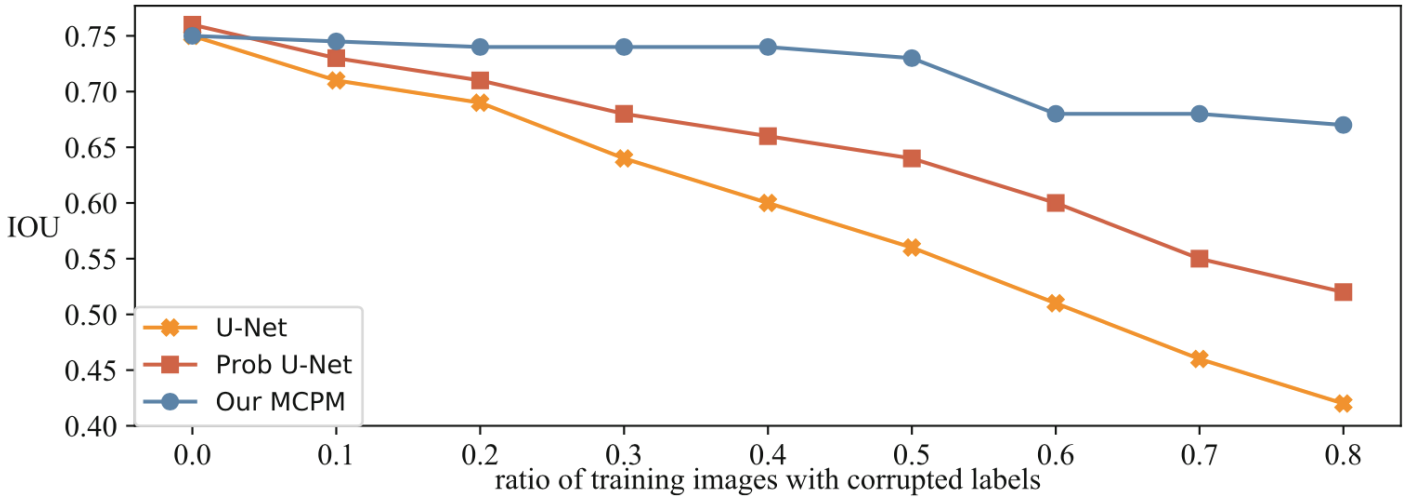
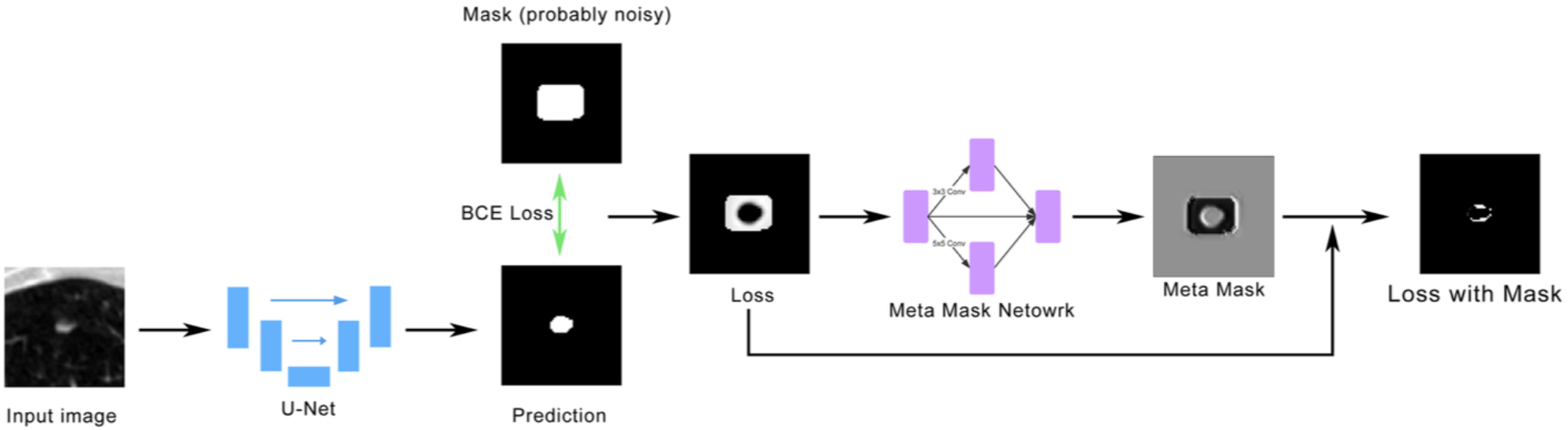



**Fig. 1.** Corrupted labels effect network's performance

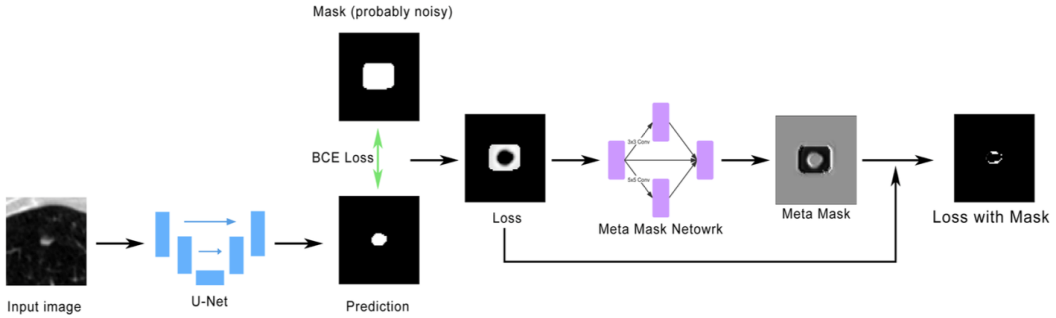
# Methodology

- Overview



# Methodology

- Optimization




Objective function:

$$\mathbf{W}^{\star}(\mathbf{\Theta}) = \arg\min_{\mathbf{W}} \frac{1}{Nhw} \sum^{N} \sum^{h} \sum^{w} \mathrm{R}^{i}_{xy} \mathrm{L}^{i}_{xy}.$$

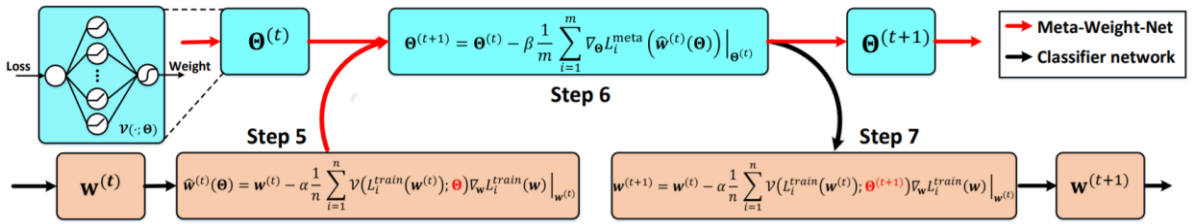
$$\mathbf{\Theta}^{*}(\mathbf{W}) = \arg\min_{\mathbf{\Theta}} \frac{1}{Mhw} \sum_{j=1}^{M} \sum_{x=1}^{h} \sum_{y=1}^{w} \hat{\mathrm{L}}^{j}_{xy}.$$

Update:

$$\mathbf{W}'^{(t)}(\mathbf{\Theta}) = \mathbf{W}^{(t)} - \alpha \frac{1}{Nhw} \sum_{i=1}^{N} \sum_{x=1}^{h} \sum_{y=1}^{w} \mathrm{R}^{i(t)}_{xy} \frac{\partial \mathrm{L}^{i}_{xy}}{\partial \mathbf{W}} \Big|_{\mathbf{W}^{(t)}}$$

$$\mathbf{\Theta}^{(t+1)} = \mathbf{\Theta}^{(t)} - \beta \frac{1}{Mhw} \sum_{j=1}^{M} \sum_{x=1}^{h} \sum_{y=1}^{w} \frac{\partial \hat{\mathrm{L}}^{j}_{xy}}{\partial \mathbf{W}'(\Theta)} \Big|_{\mathbf{W}'^{(t)}} \frac{\partial \mathbf{W}'(\Theta)}{\partial \mathbf{\Theta}} \Big|_{\mathbf{\Theta}^{(t)}}$$

$$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} - \alpha \frac{1}{Nhw} \sum_{i=1}^{N} \sum_{x=1}^{h} \sum_{y=1}^{w} \mathrm{R}^{i(t+1)}_{xy} \frac{\partial \mathrm{L}^{i}_{xy}}{\partial \mathbf{W}} \Big|_{\mathbf{W}^{(t)}}$$




**Algorithm 1** The MW-Net Learning Algorithm

**Input:** Training data $\mathcal{D}$, meta-data set $\widehat{\mathcal{D}}$, batch size $n, m$, max iterations $T$.
**Output:** Classifier network parameter $\mathbf{w}^{(T)}$

1: Initialize classifier network parameter $\mathbf{w}^{(0)}$ and Meta-Weight-Net parameter $\Theta^{(0)}$.
2: **for** $t = 0$ **to** $T - 1$ **do**
3: $\{x, y\} \leftarrow$ SampleMiniBatch($\mathcal{D}, n$).
4: $\{x^{(meta)}, y^{(meta)}\} \leftarrow$ SampleMiniBatch($\widehat{\mathcal{D}}, m$).
5: Formulate the classifier learning function $\hat{\mathbf{w}}^{(t)}(\Theta)$ by Eq. (3).
6: Update $\Theta^{(t+1)}$ by Eq. (9).
7: Update $\mathbf{w}^{(t+1)}$ by Eq. (5).
8: **end for**

*Meta-Weight-Net: Learning an Explicit Mapping For Sample Weighting (NIPS2019)

## Methodology

- Visulization

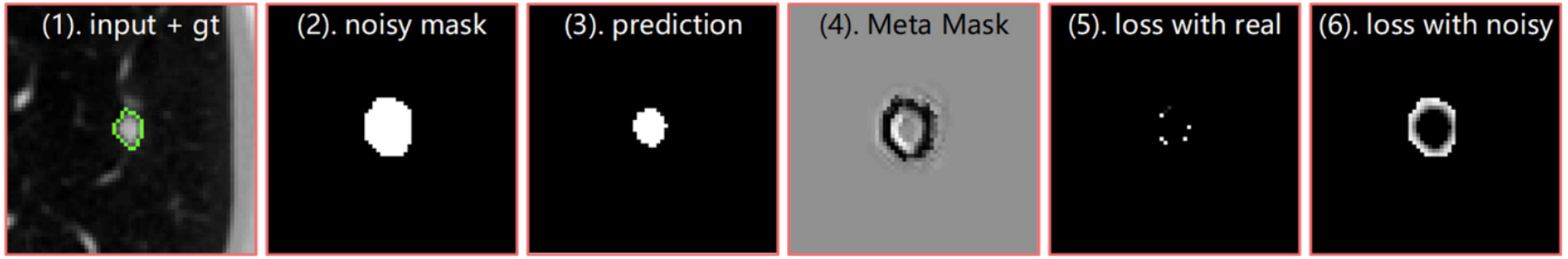



**Fig. 3.** Illustration of how meta mask network works in the training process

## Experiments

- Dataset

  LIDC-IDRI: lung CT scans, 3591 patches are cropped out, 1906 for training and 1385 images for testing, the remain 300 images are used as the meta set.

  LiTS: 130 abdomen CT scans, with annotations of liver tumors. 2214 samples are sampled, 1471, 300 and 443 images are used for training, meta weight learning, and testing respectively.

- Synthesizing noisy annotations of meta set

  a. Morphology dilation

  b. Toolkit ElasticDeform, including more complicate operations such as rotation, translation, deformation and morphology dilation

# Experiments

- Comparison with Existing Methods

**Table 1.** Results of segmentation models on LIDC-IDRI.

| Noisy | Dilation | | | ElasticDeform | | |
|---|---|---|---|---|---|---|
| Model name | mIOU | Dice | Hausdorff | mIOU | Dice | Hausdorff |
| U-Net | 62.53 | 75.56 | 1.9910 | 65.01 | 76.17 | 1.9169 |
| U-Net Meta | 60.91 | 72.21 | 2.0047 | 60.91 | 72.21 | 2.0047 |
| Prob U-Net | 66.42 | 78.39 | 1.8817 | 68.43 | 79.50 | 1.8757 |
| Phi-Seg | 67.01 | 79.06 | 1.8658 | 68.55 | 81.76 | 1.8429 |
| UA-MT | 68.18 | 80.98 | 1.8574 | 68.84 | 82.47 | 1.8523 |
| Curriculum | 67.78 | 79.54 | 1.8977 | 68.18 | 81.30 | 1.8691 |
| Few-Shot GAN | 67.74 | 78.11 | 1.9137 | 67.93 | 77.83 | 1.9223 |
| Quality Control | 65.00 | 76.50 | 1.9501 | 68.07 | 77.68 | 1.9370 |
| $U^2$ Net | 65.92 | 76.01 | 1.9666 | 67.20 | 77.05 | 1.9541 |
| MWNet | 71.56 | 81.17 | 1.7762 | 71.89 | 81.04 | 1.7680 |
| Our MCPM | **74.69** | **84.64** | **1.7198** | **75.79** | **84.99** | **1.7053** |
| U-Net Clean | 75.73 | 83.91 | 1.7051 | 75.73 | 83.91 | 1.7051 |

**Table 2.** Results (mIOU) of segmentation models using various $r$-s.

| Dataset name | LIDC-IDRI | | | | | LiTS | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $r$ | 0.8 | 0.6 | 0.4 | 0.2 | 0 | 0.8 | 0.6 | 0.4 | 0.2 | 0 |
| U-Net | 42.64 | 51.23 | 62.53 | 69.88 | 75.73 | 37.18 | 43.55 | 46.41 | 51.20 | 61.07 |
| Prob U-Net | 52.13 | 60.81 | 66.42 | 71.03 | **76.37** | 40.16 | 45.90 | 49.22 | 53.97 | 60.60 |
| MWNet | 61.28 | 67.33 | 71.56 | 72.07 | 74.40 | 43.14 | 44.97 | 51.96 | 58.65 | 59.18 |
| Our MCPM | **67.60** | **68.97** | **74.69** | **74.87** | 75.26 | **45.09** | **48.76** | **55.17** | **62.04** | **62.68** |

## Experiments

- Visualization

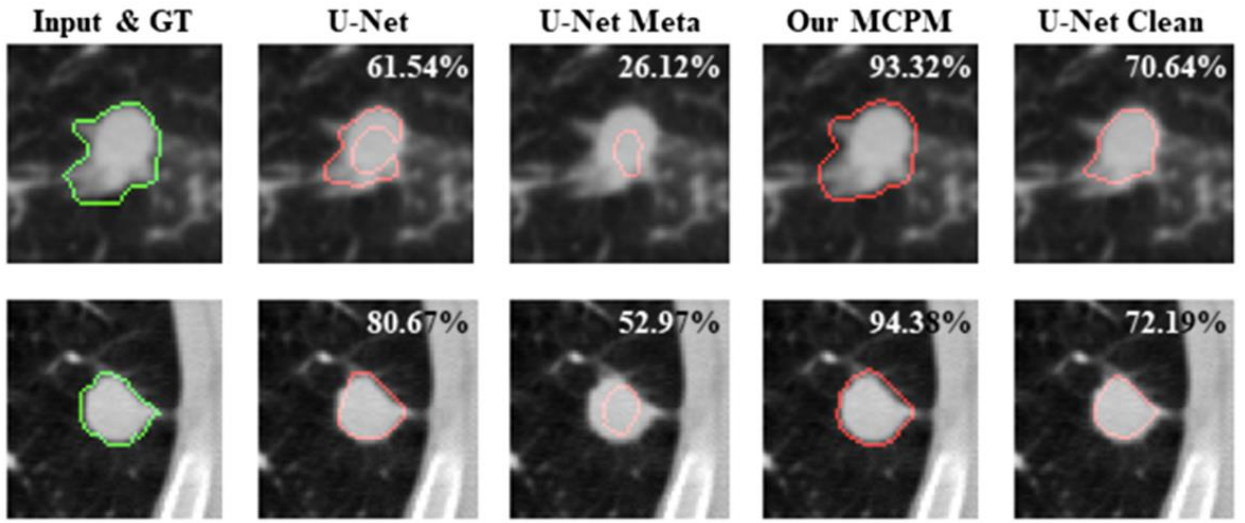



**Fig. 4.** Visualization of segmentation results. Green and red contours indicate the ground-truths and segmentation results, respectively. The Dice value is shown at the top-right corner, and our method produces much better results than other methods. (Color figure online)

(MICCAI2021)

# Distilling effective supervision for robust medical image segmentation with noisy labels

Jialin Shi[1] and Ji Wu[1,2]

[1] Department of Electronic Engineering, Tsinghua University, Beijing, China
shi-jl16@mails.tsinghua.edu.cn
[2] Institute for Precision Medicine, Tsinghua University, Beijing, China
wuji_ee@mail.tsinghua.edu.cn

# Introduction

- Related work

  pixel-wise noise estimation

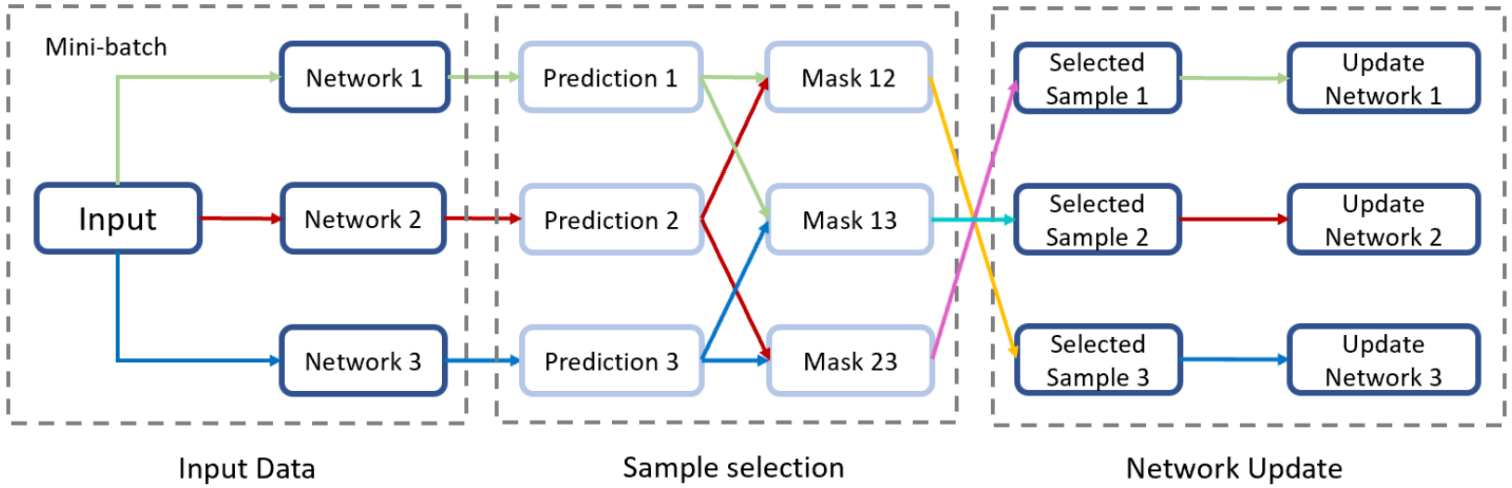
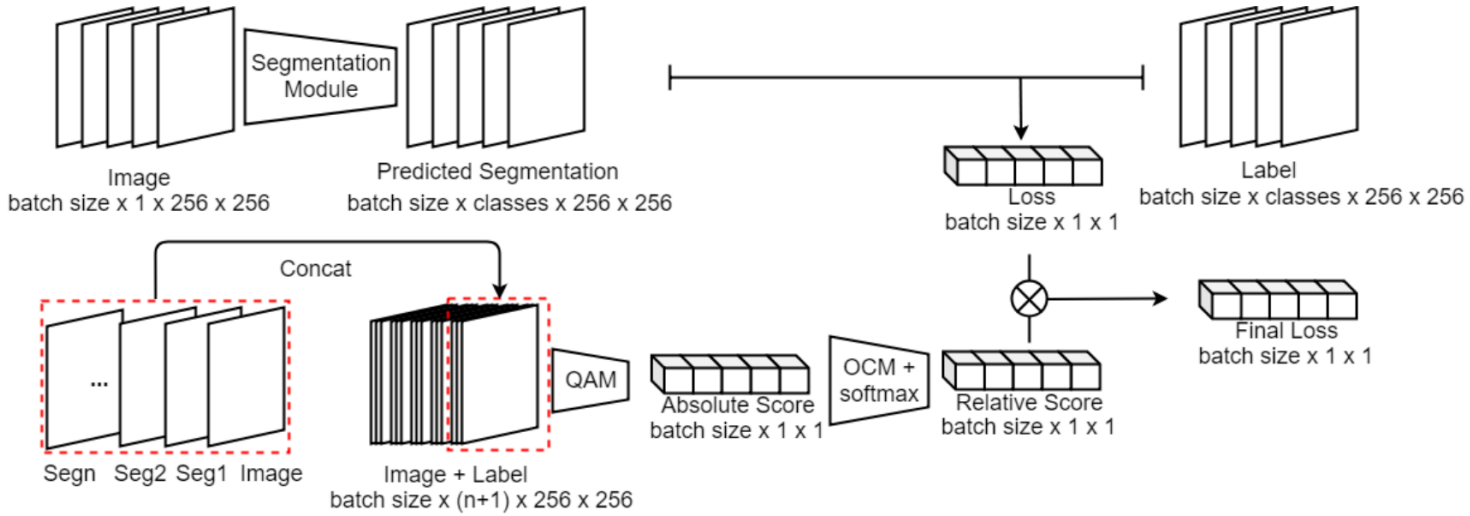



  image-level noise estimation



# Introduction

- Key Idea

  Two-phase noise estimation + learning strategy:

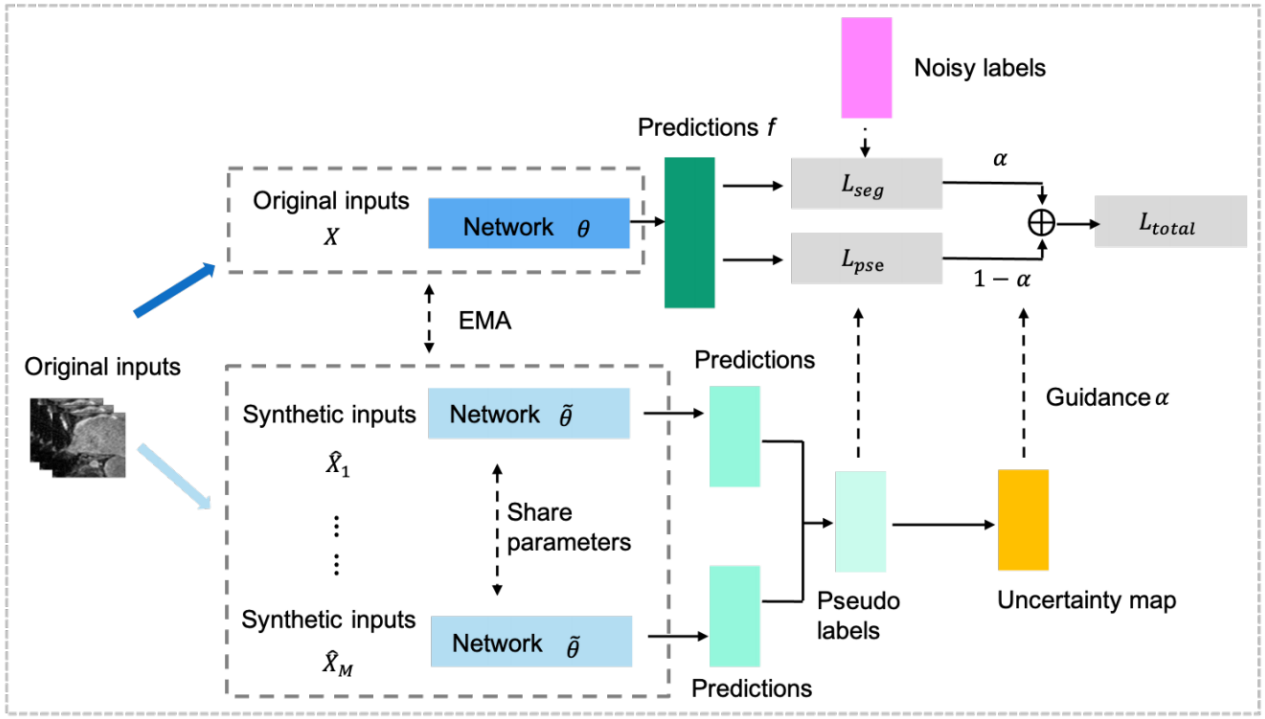
  - pixel-wise:

    intuition: the predictions under different perturbations for the same input would agreeon the relative clean labels

  - image-level

    For pixel-wise noise-tolerant learning, the pixels with high uncertainty tends to be noisy. However, there are also some clean pixels which show high uncertainty when they lie in the boundaries. If only pixel-wise robust learning is considered, the network will inevitably neglect these useful pixels.

# Methodology




auxiliary model: $\widetilde{\theta}_t = \gamma\widetilde{\theta}_{t-1} + (1-\gamma)\theta_t$

- pixel-wise:

$$u_v = \mathcal{E}[-\hat{p}_v log\hat{p}_v]$$

$$L_v = \alpha_v L_v^{seg} + (1-\alpha_v)L_v^{pse}$$

$$L_v^{seg} = \mathcal{L}_{ce}(f_v, y_v) = \mathcal{E}[-y_v log f_v]$$

$$L_v^{pse} = \mathcal{L}_{mse}(f_v, \hat{y}_v) = \mathcal{E}[||f_v - \hat{y}_v||^2]$$

- image-level:

regard every slice-level data as image-level data

$$U_i = \frac{\bar{1}}{N_i}\sum_v u_v$$

$$L_i = \alpha_i L_i^{seg} + (1-\alpha_i)L_i^{pse}$$

# Experiments

**Table 1.** Segmentation performance comparison on simulated LA noisy dataset with varying noise rates (25%, 50% and 75%). The average values (±std) over 3 repetitions are reported. The arrows indicate which direction is better.

| Method | 25% | | 50% | | 75% | |
|---|---|---|---|---|---|---|
| | Dice(%)↑ | ASD ↓ | Dice(%)↑ | ASD ↓ | Dice(%)↑ | ASD ↓ |
| V-net [16] | 86.34±0.59 | 2.72±0.36 | 82.55±0.26 | 3.35±0.01 | 72.76±1.00 | 5.48±0.06 |
| Reweighting [11] | 87.31±0.28 | 2.46±0.35 | 83.24±0.70 | 3.20±0.17 | 73.02±0.32 | 5.30±0.12 |
| Tri-network [10] | 87.92±0.44 | 2.37±0.27 | 84.79±0.44 | 2.83±0.16 | 73.88±0.46 | 5.22±0.11 |
| Pick-and-learn [13] | 88.47±0.30 | 1.92±0.24 | 85.09±0.56 | 2.73±0.20 | 73.30±0.27 | 5.11±0.08 |
| PNT | 88.29±0.43 | 1.82±0.11 | 86.16±0.69 | 2.43±0.05 | 74.92±0.19 | 5.16±0.01 |
| INT | 89.24±0.21 | 1.75±0.21 | 85.78±0.55 | 2.56±0.12 | 74.42±0.23 | 5.20±0.08 |
| PINT | **90.49±0.39** | **1.60±0.06** | **89.04±0.71** | **1.92±0.17** | **76.25±0.44** | **4.56±0.18** |

Upper bound: Dice-91.14%, ASD-1.52

**Table 2.** Segmentation performance comparison on real-world CTV dataset. The arrows indicate which direction is better and the average values (±std) over 3 repetitions are reported.

| Method | No noise [16] | V-net [16] | Re-weighting [11] | Pick-and-learn [13] | PNT | INT | PINT |
|---|---|---|---|---|---|---|---|
| Dice(%)↑ | 77.26±0.53 | 68.26±0.21 | 69.31±0.43 | 70.79±0.31 | 73.57±0.37 | 72.08±0.56 | **75.31±0.15** |
| ASD [voxel] ↓ | 1.38±0.03 | 2.25±0.02 | 2.05±0.06 | 2.11±0.07 | 1.85±0.04 | 1.92±0.08 | **1.76±0.13** |

# Experiments

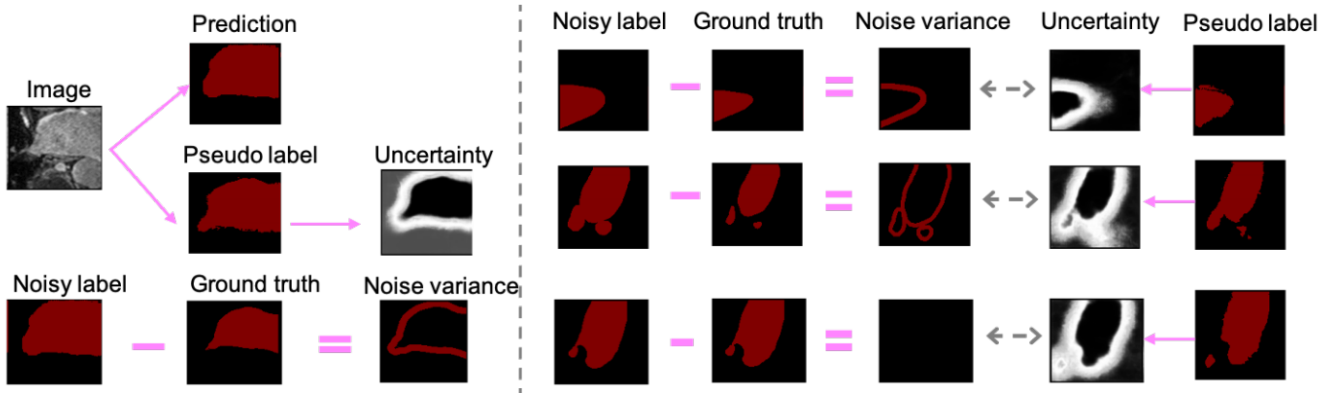



**Fig. 2.** Illustration of noise variance and pixel-wise uncertainty. The white color means higher uncertainty. The pixels with high uncertainty usually lie in the noise areas or marginal areas.

# Experiments

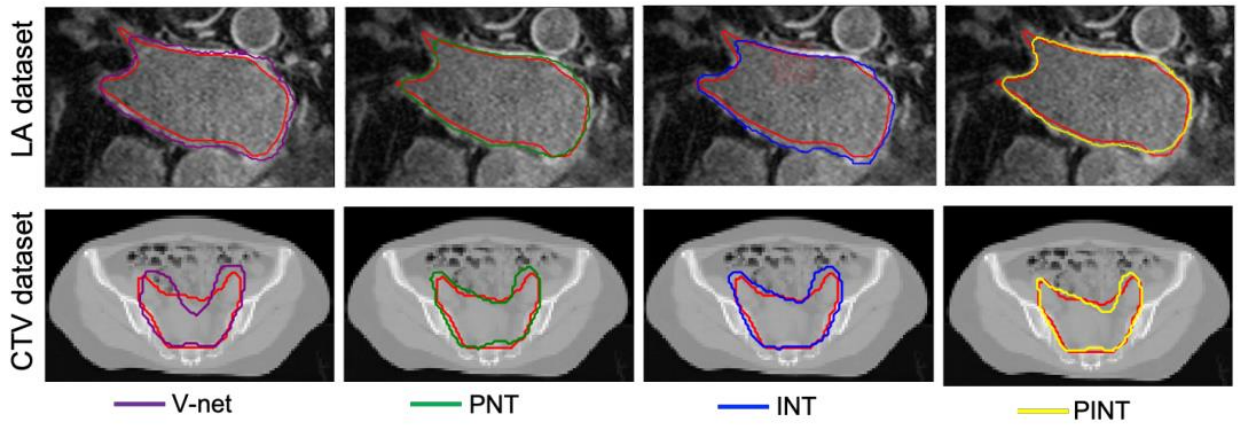



**Fig. 3.** Qualitative results of segmentation with noisy labels on simulated LA dataset and real-world CTV dataset. The ground truths, the predictions by V-net, the predictions by PNT, the predictions by INT and the predictions by PINT are colored with red, purple, green, blue and yellow, respectively.