



# HHS Public Access

Author manuscript

*IEEE Trans Med Imaging*. Author manuscript; available in PMC 2023 January 08.

Published in final edited form as:

*IEEE Trans Med Imaging*. 2022 December ; 41(12): 3952–3968. doi:10.1109/TMI.2022.3202759.

## Label Cleaning Multiple Instance Learning: Refining Coarse Annotations on Single Whole-Slide Images

**Zhenzhen Wang,**

Department of Biomedical Engineering, Mathematical Institute of Data Science, Johns Hopkins University, Baltimore, MD 21218 USA

**Carla Saoud,**

Department of Pathology, Johns Hopkins Medicine, Baltimore, MD 21218 USA

**Sintawat Wangsiricharoen,**

Department of Pathology, Johns Hopkins Medicine, Baltimore, MD 21218 USA

**Aaron W. James,**

Department of Pathology, Johns Hopkins Medicine, Baltimore, MD 21218 USA

**Aleksander S. Popel,**

Department of Biomedical Engineering, Johns Hopkins University, Baltimore, MD 21218 USA

**Jeremias Sulam [Member, IEEE]**

Department of Biomedical Engineering, Mathematical Institute of Data Science, Johns Hopkins University, Baltimore, MD 21218 USA

### Abstract

Annotating cancerous regions in whole-slide images (WSIs) of pathology samples plays a critical role in clinical diagnosis, biomedical research, and machine learning algorithms development. However, generating exhaustive and accurate annotations is labor-intensive, challenging, and costly. Drawing only coarse and approximate annotations is a much easier task, less costly, and it alleviates pathologists' workload. In this paper, we study the problem of refining these approximate annotations in digital pathology to obtain more accurate ones. Some previous works have explored obtaining machine learning models from these inaccurate annotations, but few of them tackle the refinement problem where the mislabeled regions should be explicitly identified and corrected, and all of them require a – often very large – number of training samples. We present a method, named Label Cleaning Multiple Instance Learning (LC-MIL), to refine coarse annotations on a *single* WSI without the need for external training data. Patches cropped from a WSI with inaccurate labels are processed jointly within a multiple instance learning framework, mitigating their impact on the predictive model and refining the segmentation. Our experiments on a heterogeneous WSI set with breast cancer lymph node metastasis, liver cancer, and colorectal cancer samples show that LC-MIL significantly refines the coarse annotations, out-performing state-of-the-art alternatives, even while learning from a single slide. Moreover, we demonstrate

---

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see <https://creativecommons.org/licenses/by/4.0/>

(Corresponding author: Jeremias Sulam. jsulam1@jhu.edu).

how real annotations drawn by pathologists can be efficiently refined and improved by the proposed approach. All these results demonstrate that LC-MIL is a promising, lightweight tool to provide fine-grained annotations from coarsely annotated pathology sets.

## Keywords

Whole-slide image segmentation; multiple instance learning; coarse annotations; label cleaning

---

## I. Introduction

Pathology plays a critical role in modern medicine, and particularly in cancer care. Pathology examination and diagnosis on glass slides are the gold standard for cancer diagnosis and staging. In recent years, with advances in digital scanning technology, glass slides can be digitized and stored in digital form into whole-slide images (WSIs). These WSIs contain complete tissue sections and high-level morphological details, and are changing the workflow for pathologists [1], [2].

Diagnosis by pathologists on a WSI typically include the description of cancer (*i.e.*, presence, type, and grade of cancer), the estimation of tumor size, and observation of tumor margin (whether tumor cells appear at the edge of the tissue), which is important for planning therapy and estimating prognosis. Further information, such as the detailed localization of cancer, is usually not included in the routine pathology report. However, this local information on cancer is of great interest in biomedical and pharmaceutical research. The tumor microenvironment – the stromal tissue and blood vessels surrounding the tumor cell clusters – governs the tumor growth, response to treatment, and patient prognosis [3], [4]. The quantitative analysis of the tissue microenvironment, via downstream analysis of spatial statistics and other metrics, requires a clear and accurate definition of the boundaries of the tumor, as in recent works [5], [6], [7].

On the other hand, the local detection and segmentation of histopathology images is a significant and rapidly growing field in computational pathology. Early approaches mainly focused on extracting the morphological and texture features using image processing algorithms [8], including thresholding [9], fuzzy c-means clustering [10], watershed algorithm [11], active contours [12], among others. With the advent of artificial intelligence and deep learning, a number of deep neural network models have achieved encouraging results in biomedical image segmentation [13], [14].

Whole-slide image segmentation, on the other hand, faces a unique challenge, since neural network models cannot be directly applied to the whole Gigapixel resolution WSI. A patch-based analysis (*i.e.*, training and deploying a model on numerous small patches that are cropped from the WSI) is commonly used as an alternative, and supervised deep learning has been remarkably successful when deployed this way [15]. However, high quantity (and quality) of fine-grained annotations are needed, including patch-level or pixel-level information. The latter is very costly to obtain, since detailed manual annotation on Gigapixel WSIs is extremely labor-intensive and time-consuming, and suffers from inter- and intra-observer variability [16], [17], [18]. For these reasons, state-of-art WSI

sets with detailed annotations provided by expert pathologists are very limited. The lack of large datasets with detailed and trustworthy labels is one of the biggest challenges in the development and deployment of classical supervised deep learning models in digital pathology applications.

Given the cost and difficulty of obtaining exhaustive and accurate annotations, a number of approaches attempt to address the segmentation problem in imperfect label settings [19]. Weakly-supervised learning aims to automatically infer patch-level (local) information using only slide-level (global) labels, but it typically requires thousands of WSIs as training samples [20], [21], [22]. Semi-supervised learning, on the other hand, trains models on partially annotated WSI and makes predictions for the remaining unlabeled regions, yet the partial annotations must also be conducted by experts [23]. In this paper, we re-think the WSI annotation process from a more clinically applicable scenario: Drawing coarse annotations on WSIs (e.g., rough boundaries for the cancerous regions), is much easier than detailed annotations. Such coarse annotations need only similar effort and time as the slide-level labeling, and can even be conducted by non-experts. Learning from those coarse annotations, and then refining them with computational methods, might be an efficient way to enrich the labeled pathology data with minimal effort. The resulting refined annotation can provide a more accurate “draft” for further detailed annotations by pathologists, thus alleviating their workload.

Our solution to the coarse annotation refinement problem is based on a multiple instance learning (MIL) framework that intrinsically incorporates the fact that the input annotations are imperfect, models this imperfection as patch-level label noise, and finally outputs a refined version of annotations by identifying the mislabeled patches and correcting them, as Fig. 1 illustrates. Our methodology, named Label Cleaning Multiple Instance Learning (LC-MIL), is validated in a heterogeneous set of 120 WSIs from three cancer types, including breast cancer metastasis in lymph nodes, liver cancer, and colorectal cancer, in both simulated and real experiments that demonstrate how the workload of pathologists can be alleviated. Importantly, and in order to make our method applicable to scenarios where just a few cases (slides) are available, our method can be trained and deployed on a single WSI, without the need for large training sets. Our approach significantly refines the original coarse annotations even while learning from a single WSI, and substantially outperforms state-of-the-art alternatives. To the best of our knowledge, this is the first time that multiple instance learning is employed to clean label noise, and ours is the first approach that allows for refining coarse annotations on single WSIs.

The rest of the paper is organized as follows. In Section II we provide an overview of related work. In Section III, we present the proposed methodology, LC-MIL, as well as other state-of-art baseline methods, in detail. We then proceed to our single-slide annotation refinement experiments in Section IV. After that, we provide the implementation details in Section V, and discuss the sensitivity to parameters in Section V-A. Lastly, we discuss and summarize both the implication and limitations of our approach in Section VI and Section VII.

## II. Related Work

In this section, we discuss related work from the areas that are mostly related to our contribution: label noise handling and multiple instance learning in medical imaging.

Before we describe the specific techniques and studies, we first give a brief formulation of the problem to direct the reader to the appropriate context. The coarse annotations are considered as a “noisy” label problem. More specifically, a WSI consists of a number of small patches, each of which can be assigned a label based on the annotations: patches within the annotated cancerous regions are assigned positive labels, and otherwise negative labels are assigned. However, not all of the labels are correct, given that the annotations are inaccurate. Comparing the coarse annotations with the ground truth, we will find some false positives (yellow) and false negatives (blue), as Fig. 2 illustrates. Naturally, the model has no access to the ground truth, and aims at learning from, and modifying, the coarse annotations to retrieve the correct segmentation. This is the so-called “annotation refinement” problem, interpreted as identifying mislabeled samples and correcting their labels via machine learning approaches. This is a very challenging problem: the machine learning algorithm must be able to learn from inaccurate information and over considerable heterogeneity of tissue morphology and appearance, *and* do so without an external collection of training data.

### A. Label Noise Handling

We place ourselves in a machine learning scenario, where we are given a collection of training data – pairs of samples and their corresponding labels, but part of the training samples have their labels corrupted. The learning process, predicting a rule that assigns a label to a given sample, then becomes significantly more challenging. Label noise handling is an extensively researched problem in machine learning. A large family of approaches focus on enhancing the robustness of a machine system against label noise by designing sophisticated model architectures [24], [25], [26], choosing loss functions that are tolerant to label noise [27], [28], [29], and conducting “label smoothing” [30], [31], [32]. This category of approaches typically does not evaluate the label accuracy or confidence of the training set, but tries to alleviate the impact of the corrupted labels on the performance of the model in a held-out test set.

Some other approaches attempt to evaluate the label accuracy based on predicted probabilities [33], [34], [35] or loss values [36], [37], under the intuition that samples with less confident predictions or unusually high loss values are more likely to be mislabeled. However, probability and loss cannot reveal the prediction confidence correctly in many poorly-calibrated models [38]. An auxiliary set with clean labels –if available– is popular in label noise detection, which is either used as a reference to identify potentially mislabeled samples [39], [40], or re-weigh the training samples to mitigate the impact of mislabeled samples on the system [41], [42], [43].

K-nearest neighbors (k-NN) based analysis [44], on the other hand, can be used for “editing” corrupted labels without an auxiliary clean set. The basic idea is to discard samples that are not consistent with their  $k$  nearest neighbors. In a recent work, k-NN has been

deployed within deep learning frameworks, as the deep k-NN (DkNN) approach of [45], which searches for neighbors in the feature space of a deep learning model. This method outperformed the state of the art for label noise correction, and we will revisit it in further detail as we describe our baseline in Section III-C.

## B. Multiple Instance Learning

Unlike supervised learning settings, where each training sample comes with an associated label, in multiple instance learning (MIL), one only has labels associated with groups of samples,  $X_i = \{x_k\}_{k=1}^{n_i}$ , termed *bags*, but not with the individual samples  $x_i$ , called *instances*. Although individual labels  $y_i$  exist for the instances  $x_i$ , they are unknown during training. However, the bag-level label  $Y_i$  is a function of the instance-level labels  $y_i$ . This function was simply a max pooling operator when MIL was first proposed by Dietterich *et al.* [46], denoted as  $Y = \max_i \{y_i\}$ .

Over the years, various alternative MIL formulations were developed. Ilse *et al.* [47] provide a generalization of MIL predictors as a composition of individual functions:

$$P_{Y=1} = g(\sigma(f(X))). \quad (1)$$

Here,  $f: \mathbb{R}^d \rightarrow \mathbb{H}$  is a transformation function mapping individual instances to a feature space or label space;  $\sigma: (\mathbb{H})^k \rightarrow \mathbb{H}$  is a permutation invariant pooling function that aggregates the  $k$  transformed instances within a bag; and  $g: \mathbb{H} \rightarrow \mathcal{Y}$  finally maps the aggregated instances to the corresponding bag label space.

When MIL is applied to medical imaging analysis, one typically considers an entire WSI as a bag, and regards patches cropped from the corresponding slide as instances. The bag label depends on the presence (positive) or absence (negative) of disease in the entire slide. While often successful in bag label prediction, these approaches require very large datasets with thousands of slides [20], [21], [22]. Moreover, the disease localization, or instance-level predictions, usually suffers from the lack of supervision and underperforms the fully supervised counterparts [48], [49]. Moreover, local detection is usually considered as an additional – sometimes optional – task instead of the primary goal in MIL studies. Even if a heatmap or a saliency map is generated to highlight the diagnostically significant regions, the localization performance is not always quantitatively validated [20], [21], [50].

Some works integrate other forms of weak annotations into the MIL framework to boost the local detection performance. CDWS-MIL, proposed by [51], introduced the percentage of the cancerous region within each image as an additional constraint to improve disease localization compared with using image-level labels only. CAMEL, proposed by [52], first splits the WSI into latticed patches, and considers each patch as a bag. A MIL model is then used to generate a pixel-wise heatmap for each patch. Those weak annotations, although easier to obtain than pixel-wise annotations, still need substantial effort and domain expertise. Our proposed method, on the other hand, is able to manage very coarse annotations, which are much easier to obtain, even without domain expertise.

### III. Methods

In this section, we first formulate the problem at hand and then proceed to describe our proposed approach, label cleaning multiple instance learning (LC-MIL). Finally, we describe other state-of-art methods that we use for comparison, including DkNN and Rank Pruning.

#### A. Problem Formulation

We consider a single WSI with some coarse annotations for regions of interest (e.g., cancerous regions) as a dataset with noisy labels. The WSI is latticed to generate  $N$  square patches, denoted as  $\mathbb{S} = \{(x_i, y_i)\}_{i=1}^N$ . Each patch  $x_i \in \mathbb{R}^d$  is assigned a label  $y_i \in \{0, 1\}$  based on the coarse annotations provided. To be more specific, a patch is assigned a positive label if it falls within the positively annotated area (e.g., cancerous region), and negative otherwise, as Fig. 3 illustrates. In a practical setting, there may be some patches at the intersections of two classes of regions. For those patches, the assigned labels are decided by the location of their centers.

Importantly, we do not assume that all patches in the positively annotated area are actually positive: since the coarse annotations cannot delineate the disease regions precisely, there must be patches having an incorrect label assigned to them. Similarly, not all patches in the negative annotated area are true negatives, implying that there might be “missed positive regions”. The problem of refining this coarse (and inaccurate) annotation can then be interpreted as detecting the mislabeled patches.

#### B. Label Cleaning MIL (LC-MIL)

Our proposed methodology, Label Cleaning Multiple Instance Learning (LC-MIL), tackles the noisy label problem from a MIL framework. Broadly speaking, our approach consists of a multiple instance learning model that is trained to classify *bags* of patches from a single WSI image with noisy labels. Once this model is trained, it is then employed to re-classify all patches in the image by constructing singleton bags. The corresponding predictions are used to correct the original noisy labels, and in doing so, refine the inaccurate segmentation. Our approach is general with respect to the specifics of the multiple instance model, and different specific methods are possible. In this work, we focus on an implementation of based on a deep attention mechanism [47], as well as an alternative based on neural network pooling [53]. Fig. 4 depicts an overview of our LC-MIL algorithm (in its attention-based implementation), and we now expand on the details of each component.

**1) MIL Dataset Construction:** As briefly mentioned above, in order to give our approach maximal flexibility in terms of the availability of training data, we situate ourselves in the case of having a single WSI, which represents all the data available to the algorithm – we will comment on extensions to cases with access to larger datasets later.

From a single WSI, we construct our noisy dataset of image patches  $\mathbb{S} = \{(x_i, y_i)\}_{i=1}^N$ , comprising positive and negative cases: (1)  $\mathbb{S}_P = \{(x_i, y_i) \in \mathbb{S} : y_i = 1\}$ , which consists of all of the patches with a positive label; (2)  $\mathbb{S}_N = \{(x_i, y_i) \in \mathbb{S} : y_i = 0\}$ , which consists of all of the patches with a negative label. In our context, where labels are determined by only coarse

and inaccurate annotations, there are likely false positives in  $\mathbb{S}_P$  and false negatives in  $\mathbb{S}_N$ . However, we assume that the majority of the labels in both subsets are correct. With this assumption, we construct a positive bag  $X_j$  by selecting (uniformly at random)  $n_j$  instances from  $\mathbb{S}_P$ , and construct a negative bag analogously with instances from  $\mathbb{S}_N$ . We are able to create virtually as many bags as desired from one single WSI, since the sampling is conducted with replacement. The constructed MIL dataset with  $M$  bags can be defined as:

$$S_{\text{MIL}} = \left\{ \left( X_j = \{x_{ij}\}_{i=1}^{n_j}, Y_j \right) \right\}_{j=1}^M, \quad (2)$$

where  $x_{ij}$  is the  $i$ th instance in  $X_j$ , and  $Y_j$  refers to the bag-level label, given by

$$Y_j = \begin{cases} 1, & \text{if } X_j \subset \mathbb{S}_P \\ 0, & \text{if } X_j \subset \mathbb{S}_N. \end{cases} \quad (3)$$

The number of bags  $M$ , and the number of instances within a bag  $n_j$  are adjustable parameters, which we gave the detailed setting and discussed the sensitivity of different choices in Section V-A.

**2) MIL Predictor:** We built a deep MIL model to predict the bag-level score  $P_j$  under the composite function framework as described in Section II-B and Eq. (1). Our approach is general in that it admits different forms for the functions  $g$ ,  $\sigma$ , and  $f$ , leading to different MIL methods. We focus first on a predictor similar to the one proposed in [47], composed of the three functions below:

1. A feature extractor, parameterized by a deep neural network  $f_\psi$ , is used to map the instance  $x_{ij}$  to a low-dimensional embedding, denoted as:

$$h_{ij} = f_\psi(x_{ij}). \quad (4)$$

2. An attention-based MIL pooling operator  $\sigma$ , which gives each instance a weight  $w_{ij}$  and aggregates the low-dimensional embeddings of all instances using a weighted average to generate a representation for the whole bag [47]. Importantly, these weights are learnable functions of the features, too. A softmax function is used to re-scale the weights so that they all lie in the range  $[0,1]$ , and sum up to 1. More specifically, we have that:

$$z_j = \sigma(\{h_{ij}\}_{i=1}^{n_j}) = \sum_{j=1}^{n_j} w_{ij} h_{ij}, \quad (5)$$

with weights given by

$$w_{ij} = \frac{\exp\{W \tanh(V h_{ij})\}}{\sum_{k=1}^{n_j} \exp\{W \tanh(V h_{ik})\}}. \quad (6)$$

Here, both  $W$  and  $V$  are learnable parameters.

3. A linear classifier  $g$  predicts the bag label based on the computed representation,  $z_j$ . The predicted score,  $P_j$ , is finally obtained by the appropriate logistic function, defined as:

$$P_j = g(z_j) = \frac{1}{1 + e^{-\langle g, z_j \rangle}}, \quad (7)$$

where  $g$  is a learnable vector.

Alternatively, and for the purpose of understanding the implications of the choice in the MIL predictor, we also consider a second implementation of LC-MIL by using another MIL model called mi-Net [53], whose formulation can be formally expressed as the follows.

Given a bag of instances,  $X_j = \{x_{ij}\}_{i=1}^{n_j}$ , the bag-level likelihood  $P_j$  is given by:

$$P_j = \frac{1}{n_j} \sum_{i=1}^{n_j} f(x_{ij}), \quad (8)$$

where  $f(x_{ij})$  is an instance-level classifier parameterized by a deep neural network. To differentiate these two instantiations of our approach, we name them as *LC-MIL-atten* and *LC-MIL-miNet*, respectively.

Unlike [47], who employ a cross-entropy loss, we use the focal loss [54] to train our MIL predictors in order to promote better calibration of the predicted probabilities. This loss is defined as:

$$L = -(Y_j(1 - P_j)^\gamma \log P_j + (1 - Y_j)P_j^\gamma \log(1 - P_j)). \quad (9)$$

We set the parameter  $\gamma$  as suggested by [54]. That is  $\gamma = 5$  for  $P(Y_j = 1) \in [0, 0.2)$ , and  $\gamma = 3$  for  $P(Y_j = 1) \in [0.2, 1]$ .

**3) Inference:** There is typically a “gap” between bag-level and instance-level prediction in MIL approaches, where the instance-level score is not directly predicted by the model. Although the attention pooling operator provides a way to locate key instances, those weights cannot be interpreted as instance-level scores directly. Here, we propose to use “singleton bags” as a simple solution, enabling us to exploit the (calibrated) scores provided by the classification model: for bags consisting of only one instance, the bag-level prediction is equal to the instance-level prediction. In this way, during the inference phase, we revisit the noisy dataset  $\mathbb{S} = \{(x_i, y_i)\}_{i=1}^N$  using the trained MIL model. Instead of randomly choosing a subset of instances to pack a bag, as done during training, every single instance  $x_j$  is now considered as a “single-instance bag”. Since there is only one instance in a bag, the attention-based pooling in the MIL framework has no impact during inference—in other words, the attention weights are always set to 1 for any instance—and the predicted score for each instance can be denoted as:



$$\hat{p}_i = \frac{1}{1 + e^{-\langle g, f_{\psi_2}(x_i) \rangle}}. \quad (10)$$

A predicted heatmap is then generated based on these likelihoods, where the value of each pixel refers to the risk score  $p_i \in [0, 1]$  of the corresponding patch in the WSI.

In addition to the central machine learning component of our method, detailed above, our experiments involve other implementation details, which we detail in Section V. All software implementations of our methods are publicly available.<sup>1</sup>

### C. Comparison With Other Methods

It should be noted that, to the best of our knowledge, the problem of learning a segmentation algorithm for WSI data from a single slide with inaccurate annotations has never been studied before. As a result, there are no available methods that can be deployed in an off-the-shelf manner. Thus, to provide some methods for comparison, we adapt state-of-the-art algorithms to our setting (where no auxiliary clean dataset is available).

**1) Deep k-Nearest Neighbors (DkNN) [45]:** Deep k-nearest neighbors detects label noise by using the assumption that instances within the same class should cluster together in feature space. A feature extractor parameterized by a pre-trained deep neural network is used to map instances from the input to the feature space. Then, instances having the same labels as their neighbors' are "trusted", while those having inconsistent labels with neighbors are more likely to be mislabeled. These potentially mislabeled instances are then relabeled via majority voting, which is conducted using a standard k-NN classifier.

**2) Rank Pruning [55]:** Rank Pruning identifies label noise using the predicted probabilities in a two-step process, where the instances with low probabilities are *pruned* away for the second round of training. To be more specific, a binary classifier is first fit on the noisy-labeled instances in a cross-validation manner. Each instance is then given a preliminary probability score, which is considered to reflect the prediction confidence. Certain instances are then considered as un-reliable, including those obtaining very high probability scores while their original (noisy) labels were negative, and conversely for the positive ones. After pruning such un-trusted instances, the classifier is finally re-trained on the remaining ones, and a new probability score for each instance is produced by this updated classifier trained only on trusted samples.

We also list the two implementations of LC-MIL that are used in the experiments for clarity. As we described in Section III-B, these two implementations share all the components of our LC-MIL framework, except the choice of the MIL predictor.

1. **LC-MIL-atten** uses the attention-based MIL [47] as the MIL predictor, which aggregates the embeddings of instances using an attention mechanism.

<sup>1</sup> <https://github.com/Sulam-Group/MIL-pathology>

2. **LC-MIL-miNet** uses an alternative MIL formulation called mi-Net [53], which aggregates results of the instance-level classifiers using a mean-pooling operator.

## IV. Experiments

In this section, we first describe our dataset and then show two coarse annotation refinement scenarios in simulated and real-world settings, separately.

### A. Dataset

We evaluated the coarse annotation refinement performance on three publicly available histopathology datasets. These datasets were chosen because they include different tissues and degrees of morphological heterogeneity, but also because of the availability of expert annotations that will be regarded as ground truth for the quantitative evaluation of our method.

1. **CAMELYON16** [56]: contains a total of 399 hematoxylin and eosin (H&E) stained WSIs of lymph node sections from breast cancer patients. Detailed hand-drawn contours for metastases are provided by expert pathologists.
2. **PAIP2019** [57]: contains a total of 100 H&E stained WSIs of liver cancer resection samples. The boundary of viable tumor nests was annotated precisely by expert pathologists. The viable tumor nests annotations are available for 60 WSIs.
3. **PAIP2020<sup>2</sup>**: contains a total of 118 H&E stained WSIs of colorectal cancer resection samples. The contours of the whole tumor area, which is defined as boundary enclosing dispersed viable tumor cell nests, necrosis, and peri- and intratumoral stromal tissues, are provided by expert pathologists. The whole tumor annotations are available for 47 WSIs.

The experimental design had the goal of validating the segmentation refinement capability of our method on every single slide independently, and thus slides with almost all regions of a single class (cancer or normal) were excluded to ensure that our model had sufficient positive and negative samples. We set an upper bound of 90% and a lower bound of 10% for the ratio of lesion area as the data inclusion criteria. Slides marked with “not exhaustively annotated” were also excluded. A total of 120 slides (CAMELYON16: 24; PAIP2019: 54; PAIP2020: 42) were included, as shown in Fig. 5.

### B. Synthetic Coarse Annotations

To fully evaluate the methods above, and before presenting our results in a real scenario with expert pathologists, we first study a synthetic setting where the experimental conditions (amounts of label corruptions, types of inaccurate annotations) can be easily varied and controlled. We generated coarse annotations automatically via two custom procedures,

---

<sup>2</sup>De-identified pathology images and annotations used in this research prepared and provided by the Seoul National University Hospital by a grant of the Korea Health Technology R&D Project through the Korea Health Industry Development Institute (KHIDI), funded by the Ministry of Health & Welfare, Republic of Korea (grant number: HI18C0316).

detailed in the subsections that follow. For each WSI, we considered the annotations provided by expert pathologists as the ground truth.

**1) S-I: Uniformly Flipping Patches:** A WSI with ground truth annotations provides a set of patches with correct labels. To generate simulated coarse annotations, we uniformly flip positive samples with noise rate  $\rho_1$  (fraction of flipped positive samples) and negative samples with noise rate  $\rho_0$  (fraction of flipped negative samples), allowing us to evaluate the performance of the different algorithms under a wide range of False Positives and False Negative rates. For each WSI, we generate simulated coarse annotations in this way, where  $\rho_1$  and  $\rho_0$  are sampled uniformly in the interval (0, 0.5). Fig. 6 depicts one example of these coarse annotations, and the distribution of  $\rho_0$  and  $\rho_1$  on the whole dataset.

**2) S-II: Omitting Small Lesions:** To better simulate a clinically-relevant context, we also generate noisy annotations that are similar to those provided by human annotators (note that real expert annotations will also be addressed shortly). To this end, we create coarse annotations by the following three steps: (1) Retaining only the largest cancerous region and omitting all the others; (2) Performing the morphological operation of dilation to the remaining (largest) cancerous region; (3) Taking the convex hull for the dilated lesion. As a particular case, and since almost all of the WSIs in PAIP2020 contain only one cancerous region, if there exists only one lesion this is divided in half. We also present one example of these coarse annotations and noise rate distribution on the whole dataset in Fig. 6.

These two procedures described above had the purpose of simulating both false positives as well as false negatives in the initial inaccurate annotation. The proposed method (LC-MIL) and all other competing methods were applied to each WSI independently to refine the coarse annotations. Fig. 7 presents a few examples of the refinement performance, and the reader can find more examples in Fig. 13.

In order to quantitatively evaluate how inaccurate the coarse annotations are from the true precise annotations, as well as the improvement obtained after applying our refinement method, we calculate precision (PPV), recall (TPR), and F1 scores for annotations before and after refinement. The results are summarized in Table I. Note that all the metrics were calculated per slide, and the average and standard deviation in each subset are reported. The reader can find other evaluation metrics, including specificity (TNR), negative predictive value (NPV), and intersection over Union (IoU) in Table IV and Table V. To explore how different methods perform under different noise conditions, we sort the coarse annotations according to their F1 scores, and then plot F1 scores of refined annotations in Fig. 15.

**a) Performance comparison:** The proposed method, LC-MIL, significantly improves disease localization of coarse annotations and corrects incorrect labels, and in general outperforms the competing methods based on the F1 scores. DkNN typically achieves lower metrics than Rank Pruning and LC-MIL, and only brings slight improvements for the original inaccurate annotations, which illustrates the difficulty of the learning problem under consideration. Rank Pruning generally shows stronger improvement capacity compared with DkNN, while still slightly under-performing the proposed method. Moreover, LC-MIL is especially useful in detecting missed lesions. If we focus on the values of PPV and TPR

in Table I, we will find that all the methods can efficiently improve PPV (*i.e.*, making tumor boundaries more precise), but LC-MIL shows obvious advantages in improving TPR (*i.e.*, detecting missed lesions). On the other hand, Rank Pruning generally performs best in improving PPV, and generates fewer false positives.

**b) Run-time comparison:** All methods are implemented in Pytorch and trained on a single NVIDIA GTX1080Ti GPU, and all methods make predictions on the same number of patches for each WSI. In general, LC-MIL is the fastest method. DkNN and Rank Pruning are slower: the former which needs to conduct a k-NN search on an extremely large data matrix, whereas the latter necessitates repeated training in a cross-validation manner. Note that the run-time merely depends on the number of patches of each WSI, and is not correlated with the amount of noise in the coarse annotations. We randomly select 5 slides from each subset, and run each method in exactly the same setting in order to report the per-slide run-time comparison summarized in Table II.

### C. Real-World Experiment: Improving Pathologists' Annotations

To evaluate the proposed method in a real-world scenario, we collaborate with two expert (senior) pathology residents from Johns Hopkins Medicine. Since the datasets we employ already have ground truth annotations provided by pathologists, we explore whether our proposed formulation can be used to improve the pathologists' workflow by refining approximate segmentations. In particular, pathologists were asked to provide segmentations of the tumor regions in each of the 120 WSIs, but to do so in a time-constrained manner, with 30 seconds per slide. We regard these as the coarse annotations to refine.

The proposed LC-MIL approach, as well as other competing methods, were applied in order to refine these quick and coarse annotations. Note that the refinement was still conducted on each slide and each pathologist's annotations independently. Fig. 8 presents some obtained examples, and the overall refinement results are summarized in Table III. The reader can also find more evaluation metrics in Table VI and more examples in Fig. 14. We also show the distributions of coarse annotations drawn by pathologists in a time-constrained manner, as well as the refinement performance in different noise conditions, as Fig. 16 shows.

As Table III indicates, all methods improve the coarse annotations to some extent on all three datasets. LC-MIL achieves the best overall performance, obtaining the second-best F1 only on PAIP2020. Interestingly, the best method for refining coarse annotations on colorectal samples (PAIP2020) is DkNN, which achieved the lowest F1 score in the synthetic datasets. We believe that this is mainly caused by the heterogeneity and complexity of colon histology: Colon tissue samples typically contain loose connective tissue, smooth muscle, and epithelial tissue, where cancer begins developing. The texture difference between benign and malignant epithelial tissue is much more subtle than the dissimilarity across different tissue types. The LC-MIL model seems to be more easily misled to discriminate epithelial tissues from any other regions, resulting in some non-cancerous epithelial tissue being detected as (false) positive. On the other hand, when pathologists annotate colorectal cancer, they make use of both local (*e.g.*, texture) and global (*e.g.*, location, shape) information, and thus rarely misclassify normal epithelial tissues. It is

worth noting that pathologists' annotations are sufficiently good on those samples, rarely missing lesions (i.e. they have high TPR), and the only imperfection resides on rough boundaries. DkNN seems to be particularly suitable for this condition, as it might be the most conservative among all studied methods—keeping the overall segmentation while making only local improvements on boundaries—and explaining the difference in observed performance. More broadly, handling tissue samples with confounding morphology can be a challenging problem, and we expand this discussion in Section VI.

## V. Implementation Details and Parameter Sensitivity

In addition to the central machine learning component of our method, detailed above in Section III, these experiments involve other implementation details common in image analysis pipelines, which we detail next.

### Tissue Region Identification:

For each digitized slide, our pipeline begins with the automatic detection of tissue regions to exclude irrelevant (e.g. blank) sections. Gigapixel WSIs are first loaded into memory at a down-sampled resolution (e.g.,  $256\times$  downscale), which we use for the detection of empty regions. The detection methods vary slightly for different data sources. For CAMELYON16, the downsampled thumbnail is converted from RGB to the HSV color space. Otsu's algorithm [58] is applied to the H and S channels independently and then two masks are combined (by taking their intersection) to generate the final binary mask. For PAIP2019 and PAIP2020, the tissue masks are generated by applying the RGB thresholds (235, 210, 235) on each image, as suggested by the respective data provider. We present some examples of the tissue masks in Fig. 9.

### WSI Patching:

The size of each patch is set to  $256 \times 256$  (pixels). Technically, every pixel can be the center of a unique patch, resulting in millions of patches that highly overlap with each other. In this way, the predicted heatmaps, or lesion maps, have the same size and resolution as the original WSIs. However, such an approach is extremely computationally expensive and time-consuming, turning it prohibitive. In our experiments, we extracted patches with no overlap in WSIs scanned at  $\times 40$  magnification (CAMELYON16 and PAIP2020). For slides scanned at  $\times 20$  magnification (PAIP2019), cropped patches have 75% overlap with neighboring patches. The size and overlap of cropped patches naturally impact the resolution of the predicted heatmaps. For CAMELYON16 and PAIP2020, the predicted heatmaps are  $256\times$  downsampled from the original WSI; for PAIP2019, the predicted heatmaps are  $128\times$  downsampled from the original resolution. Our overall approach is not limited to these choices, and could be applicable to other settings too.

### Neural Networks Architecture:

The feature extractors used in LC-MIL ( $f\psi$ ) are built based on the 16-layer VGGNet architecture [59]. The last FCN layer is removed to generate  $f\psi$ . Since the training data is limited (one WSI), the VGGNet used has been pre-trained on the ImageNet dataset

[60]. When they are further fine-tuned on the histopathology images, the parameters in the convolutional layers are kept fixed.

### Learning Hyper-Parameters

The MIL model used in LC-MIL is trained on 1,000 MIL bags, consisting of 500 positive and 500 negative bags. Adam optimizer [61] is used with an initial learning rate of  $5 \times 10^{-5}$ , and the learning rate decays 50% every 100 bags. For CAMELYON16 and PAIP2019, each MIL bag contains 10 instances; for PAIP2020, each bag contains 3 instances. We discuss the sensitivity to the choice of number of bags and bag size below.

### Post-Processing

To obtain a fair comparison between the annotations before and after refinement, we conduct some simple post-processing on the heatmaps generated by either DkNN (binary map), Rank Pruning (scalar map), or LC-MIL (scalar map). The binary heatmap predicted by DkNN is post-processed by simple morphology operations: both small holes (smaller than 100 pixels) and small objects (smaller than 100 pixels) are removed. The scalar heatmap predicted by Rank Pruning and LC-MIL is first converted to a binary map using thresholding. The threshold  $v_0$  is decided by applying Otsu's algorithm [58] to the predicted scores of instances that are originally (coarsely) annotated positive, denoted as  $v_0 = \text{OTSU}(\{p_i^+ x_i \in S_p\})$ . Finally, the same morphology operations conducted for DkNN are applied to these binary maps.

#### A. Sensitivity to Parameters

**1) Using Multiple Slides:** A natural extension of the proposed method is to incorporate multiple slides for training when a larger dataset is available. This can be implemented by slightly modifying the LC-MIL framework described in Section III-B. To be more specific, MIL bags can be constructed from every single slide in exactly the same way, as described in Eq. 2. After that, these MIL bags from different slides can be combined to form a larger MIL dataset, denoted as  $S'_{\text{MIL}} = \{S_{\text{MIL}}^{(0)}, \dots, S_{\text{MIL}}^{(k)}\}$ , where  $k$  is the number of slides. The MIL model is then trained on this larger MIL dataset  $S'_{\text{MIL}}$ . At inference stage, the trained MIL model is used to predict tumor segmentation on held-out slides.

We empirically tested this new implementation and explored how the number of slides affects the prediction performance. The detailed setting is described as follows. We randomly split the 54 WSIs in PAIP2019 into a training set (49 slides), and a held-out validation set (5 slides). The MIL model is trained on  $k$  slides (with coarse annotations) that are randomly selected from the training set, and tested on the held-out validation set. We set  $k = 1, 2, 4, 8$  to explore different conditions. At each  $k$ , the training-validation process is repeated 5. We plot the F1 scores of the prediction results on the held-out validation set in Fig. 10. As can be seen, as  $k$  increases the obtained F1 scores of the predicted annotations increase, and their variance decreases. This experiment demonstrates how the proposed LC-MIL framework could be adjusted in a multi-slides setting and make predictions on held-out sets containing slides with no annotations. Naturally, this experiment also demonstrates that a larger training set leads to better and more stable predictions.

**2) Choice of Bag Size and Bag Numbers:** As presented in Eq. 2, there are two adjustable hyper-parameters in the construction of MIL sets: the number of instances within a bag (i.e. bag size), denoted as  $n_j$ , and the number of bags, denoted as  $M$ . Here we present the detailed settings for those parameters, and discuss the sensitivity to their respective choices. **Bag size  $n_j$ .** The bag size is affected by the amount of label noise as well as the complexity of the task (in terms of the heterogeneity of the histopathology images), and both aspects should be considered in determining an optimal bag size. We first conduct an experiment to discuss the choice of bag size in different noise conditions. We randomly select 4 slides, and generate three versions of coarse annotations for each of them. To explicitly control the noise rates, those coarse annotations are generated by uniformly flipping patches, as we described in Section IV-B.1. The noise rates for these three versions of coarse annotations are: (1)  $\rho_0 = \rho_1 = 10\%$ ; (2)  $\rho_0 = \rho_1 = 20\%$ ; (3)  $\rho_0 = \rho_1 = 40\%$ . For each slide with a specific version of coarse annotations, we applied LC-MIL-atten to refine the annotations, and the mean F1 scores of refined annotations are summarized in Fig. 11. The presented results suggest that a bag size around 10 can be considered an appropriate setting in general. When noise rate is low (e.g.,  $\rho_0 = \rho_1 = 10\%$ ), a smaller bag size achieves the best results. This is consistent with intuition—in the extreme case where coarse annotations are exactly the ground truth (i.e. no label noise exists), a bag size of 1 should be the best choice. Naturally, this case reduces the MIL classifier to a standard instance-level binary classifier. On the other hand, we do not suggest a very large bag size (e.g.,  $n_j > 50$ ) in all circumstances. Although a large bag size guarantees that the majority of instances within a MIL bag are correctly labeled, it also brings substantial ambiguity to instance-level prediction, and might decrease the F1 scores of patch-wise classification.

Furthermore, we recommend using a smaller bag size for tissues with compounded morphology, such as colorectal tissues. In the colorectal slides, normal epithelial tissue shows similar textures to cancerous tissues, while other components (fat tissue, smooth muscle) have significantly different textures, simply because they are different types of tissues. When humans identify tumors in colorectal samples, both local texture and global information are taken into account. On the other hand, LC-MIL can purely utilize local texture information, and tend to misclassify normal epithelial tissues as tumor, especially when the bag size is large. Utilizing a smaller bag size in this case leads to a more conservative refinement system, and thus decreases false negatives.

In this work, we used  $n_j = 10$  for slides in CAMELYON16 (lymph node metastasis of breast cancer) and PAIP2019 (liver cancer), and  $n_j = 3$  for slides in PAIP2020 (colorectal cancer) considering its complex and heterogeneous morphology. Note however that these parameters were not optimized per case. Further improvements are to be expected if these parameters are adjusted based on the label noise conditions and specific histology in the WSI. We leave this as a future direction of the proposed method.

**a) Bag number  $M$ :** The number of bags represents the number of training samples provided to the multiple instance learner. Consequently, a minimal number of  $M$  is expected for the algorithm to achieve a reasonable performance. We will now demonstrate that this number is relatively small, so that in all cases the models have sufficient training data. To explore the effect of bag number  $M$ , we use a constant bag size of 10, and gradually increase

$M$ . We used the same slides and coarse annotations as those used in the discussion of bag size  $n_j$ . The F1 scores of refined annotations are summarized in Fig. 12, demonstrating that 400 bags suffice to obtain reasonable results in the single-slide refinement setting. In this work, we set  $M=1000$  for every single slide in all experiments.

## VI. Discussion

The main hypothesis studied in this work is that coarse annotations on whole-slide images can be refined automatically even from very limited data, alleviating the workload of expert pathologists. This problem had remained unstudied until now due to the technical challenges that often make deep learning models require very large training sets.

To test our hypothesis, we developed the Label Cleaning Multiple Instance Learning (LC-MIL) method to refine the coarse annotations. Our experiments on a heterogeneous dataset with 120 WSIs across three different types of cancers show that LC-MIL can be used to generate a significantly better version of disease segmentation, even while learning from a single WSI with very coarse annotations. When compared to other baseline alternatives specifically adapted to this setting (since again, this problem had not been studied before), our proposed algorithm generally outperforms competing methods across all cases considered.

Our methodology has implications in both engineering and clinical domains. From a machine learning perspective, LC-MIL demonstrates the potential of the MIL framework to be used in a label cleaning context. Our setting can be considered as a significant complement to existing label cleaning methods, without requiring the need for an auxiliary clean set of samples. From a clinical perspective, the proposed methodology could be used to alleviate pathologists' workload in annotation tasks. LC-MIL allows pathologists to draw coarse annotations quickly and obtain a refined version from it. On the other hand, our results show that the refinement produced by LC-MIL is especially significant when the coarse annotations are very inaccurate, which implies that the proposed methodology could be useful for helping very inexperienced annotators. Importantly, our approach is particularly useful for increasing TPR, therefore detecting tumor areas that might have been missed by an inexperienced, or distracted, annotator.

A significant advantage of the LC-MIL is its ability to work in extreme data-scarcity scenarios, as it can be deployed on a single WSI. We have chosen to study this setting to showcase the flexibility of our approach. However, this also constrains the minimal and maximal ratio of cancerous regions if the refinement is conducted per slide, since a minimal amount of both negative and positive samples within one slide are needed for the learning algorithm. This limitation could be addressed by moving beyond the *single* WSI case and aggregating patches from multiple slides during the learning phase, if such data is available. Moving from this single-slide to multiple-slides settings is a natural extension of our methodology, as we demonstrated in Section V-A.1.

Another limitation of LC-MIL, also shared by other binary classifiers, is that the model can be confused by irrelevant information when the histology structure is complex and



heterogeneous. An example of this is represented by the difficulty in discriminating epithelial tissue from other types of tissues (loose connective tissue, smooth muscle), instead of focusing on learning the difference between benign and malignant tissue. On the other hand, this may not be a challenging task for a human (experienced) annotator. This could be addressed by extending the current version of LC-MIL to a multi-class setting. Moreover, pathologists usually make use of both local texture information and global information (e.g., shape, location) to make decisions, while our proposed approach, as well as other patch-based methods, merely depend on the local texture and fail to incorporate global information. Efficiently incorporating this global context can be a challenging and significant problem, which we also leave as future work.

An important open question remains: can detailed and labor-intensive annotations be replaced by simple and quick coarse annotations, with the help of our methodology? In our real-world experiment, where expert pathologists made quick annotations that were later refined by our LC-MIL algorithm, about 0.85 F1 scores were obtained (from initial values of about 0.73, an improvement of 16%), having taken only 30 seconds for each slide. Moreover, it remains unclear what the inter- and intra-reader variability of these annotations are. Judging by other studies in different contexts [17], [18], it is natural to assume that an F1 score of 1 is unrealistic across readers, and that F1 scores in the range of 0.9–1 might suffice. Further and larger scale studies should be designed to thoroughly evaluate this possibility.

## VII. Conclusion

Altogether, in this paper we developed LC-MIL, a label cleaning method under a multiple instance learning framework, to automatically refine coarse annotations on a single WSI. The proposed methodology demonstrates the potential of the MIL framework in a label cleaning context, and provides a new way of deploying MIL models so that they can be trained on even a single slide. LC-MIL holds promise to relieve the workload of pathologists, as well as in helping inexperienced annotators with challenging annotation tasks.

## Acknowledgment

The authors would like to thank Dr. Ashley M. Cimino-Mathews and Dr. Marissa Janine White from the Department of Pathology, School of Medicine, Johns Hopkins University, for their valuable advice and insights in histopathology. They also thank Dr. Adam Charles, Haoyang Mi, and Jacopo Teneggi from the Department of Biomedical Engineering, Johns Hopkins University, for their useful advice and discussions, as well as the anonymous reviewers for their constructive comments.

This work was supported in part by CISCO Research under Grant CG2686384 and in part by NIH under Grant R01CA138264 and Grant U01CA212007.

## Appendix

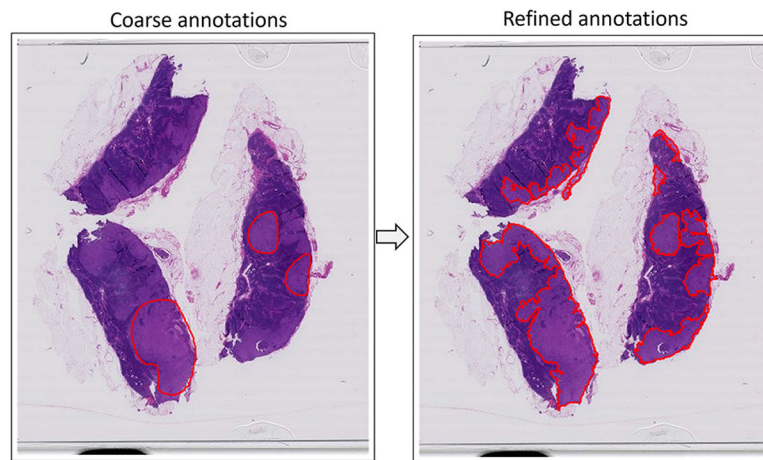
See Tables IV–VI and Figures 13–16.

## References

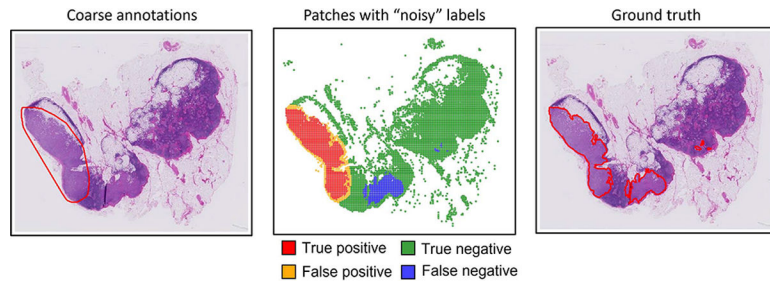
- [1]. Hanna MG, Parwani A, and Sirintrapun SJ, "Whole slide imaging: Technology and applications," *Adv. Anatomic Pathol*, vol. 27, no. 4, pp. 251–259, 2020.
- [2]. Melo RCN, Raas MWD, Palazzi C, Neves VH, Malta KK, and Silva TP, "Whole slide imaging and its applications to histopathological studies of liver disorders," *Frontiers Med*, vol. 6, p. 310, Jan. 2020.
- [3]. Song J et al. , "Hepatic stellate cells activated by acidic tumor microenvironment promote the metastasis of hepatocellular carcinoma via osteopontin," *Cancer Lett*, vol. 356, no. 2, pp. 713–720, Jan. 2015. [PubMed: 25449435]
- [4]. Joyce JA and Fearon DT, "T cell exclusion, immune privilege, and the tumor microenvironment," *Science*, vol. 348, no. 6230, pp. 74–80, Apr. 2015. [PubMed: 25838376]
- [5]. Mi H et al. , "Digital pathology analysis quantifies spatial heterogeneity of CD3, CD4, CD8, CD20, and FoxP3 immune markers in triple-negative breast cancer," *Frontiers Physiol.*, vol. 11, Oct. 2020, Art. no. 583333.
- [6]. Gong C et al. , "Quantitative characterization of CD8+ t cell clustering and spatial heterogeneity in solid tumors," *Frontiers Oncol*, vol. 8, p. 649, Jan. 2019.
- [7]. Schwen LO et al. , "Data-driven discovery of immune contexture biomarkers," *Frontiers Oncol*, vol. 8, p. 627, Dec. 2018.
- [8]. Gurcan MN, Boucheron LE, Can A, Madabhushi A, Rajpoot NM, and Yener B, "Histopathological image analysis: A review," *IEEE Rev. Biomed. Eng*, vol. 2, pp. 147–171, 2009. [PubMed: 20671804]
- [9]. Petushi S, Garcia FU, Haber MM, Katsinis C, and Tozeren A, "Large-scale computations on histology images reveal grade-differentiating parameters for breast cancer," *BMC Med. Imag*, vol. 6, no. 1, pp. 1–11, Apr. 2006.
- [10]. Kande GB, Subbaiah PV, and Savithri TS, "Unsupervised fuzzy based vessel segmentation in pathological digital fundus images," *J. Med. Syst*, vol. 34, pp. 849–858, Oct. 2009. [PubMed: 20703624]
- [11]. Veta M, van Diest PJ, Kornegoor R, Huisman A, Viergever MA, and Pluim JPW, "Automatic nuclei segmentation in H&E stained breast cancer histopathology images," *PLoS ONE*, vol. 8, no. 7, Jul. 2013, Art. no. e70221.
- [12]. Song T-H, Sanchez V, ElDaly H, and Rajpoot NM, "Dual-channel active contour model for megakaryocytic cell segmentation in bone marrow trephine histology images," *IEEE Trans. Biomed. Eng*, vol. 64, no. 12, pp. 2913–2923, Dec. 2017. [PubMed: 28391186]
- [13]. Ronneberger O, Fischer P, and Brox T, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. MICCAI*, Munich, Germany, 2015, pp. 234–241.
- [14]. Siddique N, Paheding S, Elkin CP, and Devabhaktuni V, "U-Net and its variants for medical image segmentation: A review of theory and applications," *IEEE Access*, vol. 9, pp. 82031–82057, 2021.
- [15]. Wang S, Yang DM, Rong R, Zhan X, and Xiao G, "Pathology image analysis using segmentation deep learning algorithms," *Amer. J. Pathol*, vol. 189, no. 9, pp. 1686–1698, Sep. 2019. [PubMed: 31199919]
- [16]. Longacre TA et al. , "Interobserver agreement and reproducibility in classification of invasive breast carcinoma: An NCI breast cancer family registry study," *Modern Pathol*, vol. 19, no. 2, pp. 195–207, Feb. 2006.
- [17]. Foss FA, Milkins S, and McGregor AH, "Inter-observer variability in the histological assessment of colorectal polyps detected through the NHS bowel cancer screening programme," *Histopathology*, vol. 61, no. 1, pp. 47–52, Jul. 2012. [PubMed: 22486166]
- [18]. van den Einden LC et al. , "Interobserver variability and the effect of education in the histopathological diagnosis of differentiated vulvar intraepithelial neoplasia," *Modern Pathol*, vol. 26, no. 6, pp. 874–880, Jun. 2013.
- [19]. Cheplygina V, de Bruijne M, and Pluim JP, "Not-so-supervised: A survey of semi-supervised, multi-instance, and transfer learning in medical image analysis," *Med. Image Anal*, vol. 54, pp. 280–296, May 2019. [PubMed: 30959445]

- [20]. Campanella G et al. , “Clinical-grade computational pathology using weakly supervised deep learning on whole slide images,” *Nature Med*, vol. 25, no. 8, pp. 1301–1309, Aug. 2020.
- [21]. Lu MY, Williamson DFK, Chen TY, Chen RJ, Barbieri M, and Mahmood F, “Data-efficient and weakly supervised computational pathology on whole-slide images,” *Nature Biomed. Eng*, vol. 5, no. 6, pp. 555–570, Jun. 2021. [PubMed: 33649564]
- [22]. Srinidhi CL, Ciga O, and Martel AL, “Deep neural network models for computational histopathology: A survey,” *Med. Image Anal*, vol. 67, Jan. 2021, Art. no. 101813.
- [23]. Cheng H-T et al., “Self-similarity student for partial label histopathology image segmentation,” in *Proc. ECCV, Glasgow, U.K., 2020*, pp. 117–132.
- [24]. Sukhbaatar S, Bruna J, Paluri M, Bourdev L, and Fergus R, “Training convolutional networks with noisy labels,” 2014, arXiv:1406.2080.
- [25]. Goldberger J and Ben-Reuven E, “Training deep neural-networks using a noise adaptation layer,” in *Proc. ICLR, Toulon, France, 2017*.
- [26]. Dgani Y, Greenspan H, and Goldberger J, “Training a neural network based on unreliable human annotation of medical images,” in *Proc. IEEE 15th Int. Symp. Biomed. Imag. (ISBI), Washington, DC, USA, Apr. 2018*, pp. 39–42.
- [27]. Ghosh A, Kumar H, and Sastry P, “Robust loss functions under label noise for deep neural networks,” in *Proc. AAAI Conf. Artif. Intell., San Francisco, CA, USA, 2017*, vol. 31, no. 1.
- [28]. Matuszewski DJ and Sintorn I-M, “Minimal annotation training for segmentation of microscopy images,” in *Proc. IEEE 15th Int. Symp. Biomed. Imag. (ISBI), Washington, DC, USA, Apr. 2018*, pp. 387–390.
- [29]. Rister B, Yi D, Shivakumar K, Nobashi T, and Rubin DL, “CT organ segmentation using GPU data augmentation, unsupervised labels and IOU loss,” 2018, arXiv:1811.11226.
- [30]. Gao B-B, Xing C, Xie C-W, Wu J, and Geng X, “Deep label distribution learning with label ambiguity,” *IEEE Trans. Image Process*, vol. 26, no. 6, pp. 2825–2838, Jun. 2017. [PubMed: 28371776]
- [31]. Pham HH, Le TT, Tran DQ, Ngo DT, and Nguyen HQ, “Interpreting chest X-rays via CNNs that exploit hierarchical disease dependencies and uncertainty labels,” *Neurocomputing*, vol. 437, pp. 186–194, May 2021.
- [32]. Islam M and Glocker B, “Spatially varying label smoothing: Capturing uncertainty from expert annotations,” in *Proc. IPMI, 2021*, pp. 677–688.
- [33]. Northcutt CG, Wu T, and Chuang IL, “Learning with confident examples: Rank pruning for robust classification with noisy labels,” in *Proc. UAI, Sydney, NSW, Australia, 2017*, pp. 1321–1330.
- [34]. Ding Y, Wang L, Fan D, and Gong B, “A semi-supervised two-stage approach to learning from noisy labels,” in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV), Lake Tahoe, NV, USA, Mar. 2018*, pp. 1215–1224.
- [35]. Köhler JM, Autenrieth M, and Beluch WH, “Uncertainty based detection and relabeling of noisy image labels,” presented at the *CVPR Workshops, 2019*.
- [36]. Wang X, Hua Y, Kodirov E, and Robertson N, “Emphasis regularisation by gradient rescaling for training deep neural networks with noisy labels,” 2019, arXiv:1905.11233.
- [37]. Shu J et al., “Meta-weight-net: Learning an explicit mapping for sample weighting,” in *Proc. NeurIPS, Vancouver, BC, USA, 2019*, pp. 1917–1928.
- [38]. Guo C and Pleiss G, “On calibration of modern neural networks,” in *Proc. Int. Conf. Mach. Learn., Sydney, NSW, Australia, 2017*, pp. 1321–1330.
- [39]. Vo PD, Ginsca A, Le Borgne H, and Popescu A, “Effective training of convolutional networks using noisy web images,” in *Proc. 13th Int. Workshop Content-Based Multimedia Indexing (CBMI), Prague, Czech, Jun. 2015*, pp. 1–6.
- [40]. Lee K-H, He X, Zhang L, and Yang L, “CleanNet: Transfer learning for scalable image classifier training with label noise,” in *Proc. CVPR, Salt Lake City, UT, USA, 2018*, pp. 5447–5456.
- [41]. Ren M, Zeng W, Yang B, and Urtasun R, “Learning to reweight examples for robust deep learning,” in *Proc. ICML, Stockholm, Sweden, 2018*, pp. 4334–4343.

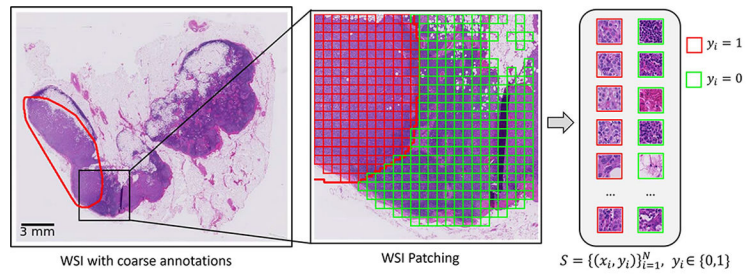
- [42]. Le H, Samaras D, Kurc T, Gupta R, Shroyer K, and Saltz J, “Pancreatic cancer detection in whole slide images using noisy label annotations,” in Proc. MICCAI, Shenzhen, China, 2019, pp. 541–549.
- [43]. Mirikharaji Z, Yan Y, and Hamarneh G, “Learning to segment skin lesions from noisy annotations,” in Domain Adaptation and Representation Transfer and Medical Image Learning with Less Labels and Imperfect Data. Cham, Switzerland: Springer, 2019, pp. 207–215.
- [44]. Wilson DL, “Asymptotic properties of nearest neighbor rules using edited data,” IEEE Trans. Syst., Man, Cybern, vol. SMC-2, no. 3, pp. 408–421, Jul. 1972.
- [45]. Bahri D, Jiang H, and Gupta M, “Deep k-NN for noisy labels,” in Proc. ICML, 2020, pp. 540–550.
- [46]. Dietterich TG, Lathrop RH, and Lozano-Pérez T, “Solving the multiple instance problem with axis-parallel rectangles,” Artif. Intell, vol. 89, nos. 1–2, pp. 31–71, Jan. 1997.
- [47]. Ilse M, Tomczak J, and Welling M, “Attention-based deep multiple instance learning,” in Proc. ICML, Stockholm, Sweden, 2018, pp. 2127–2136.
- [48]. Xu Y, Zhu J-Y, Chang EI-C, Lai M, and Tu Z, “Weakly supervised histopathology cancer image segmentation and classification,” Med. image Anal, vol. 18, no. 3, pp. 591–604, 2014. [PubMed: 24637156]
- [49]. Courtiol P, Tramel EW, Sanselme M, and Wainrib G, “Classification and disease localization in histopathology using only global labels: A weakly-supervised approach,” 2018, arXiv:1802.02212.
- [50]. Wang X, Yan Y, Tang P, Bai X, and Liu W, “Revisiting multiple instance neural networks,” Pattern Recognit, vol. 74, pp. 15–24, Feb. 2018.
- [51]. Jia Z, Huang X, Chang EI-C and Xu Y, “Constrained deep weak supervision for histopathology image segmentation,” IEEE Trans. Med. Imag, vol. 36, no. 11, pp. 2376–2388, Nov. 2017.
- [52]. Xu G et al., “CAMEL: A weakly supervised learning framework for histopathology image segmentation,” in Proc. ICCV, Seoul, South Korea, 2019, pp. 10682–10691.
- [53]. Wang X, Yan Y, Tang P, Bai X, and Liu W, “Revisiting multiple instance neural networks,” Pattern Recognit, vol. 74, pp. 15–24, Feb. 2018.
- [54]. Mukhoti J, Kulharia V, Sanyal A, Golodetz S, Torr PHS, and Dokania PK, “Calibrating deep neural networks using focal loss,” 2020, arXiv:2002.09437.
- [55]. Northcutt CG, Wu T, and Chuang IL, “Learning with confident examples: Rank pruning for robust classification with noisy labels,” 2017, arXiv:1705.01936.
- [56]. Litjens G et al. , “1399 H&E-stained sentinel lymph node sections of breast cancer patients: The CAMELYON dataset,” GigaScience, vol. 7, no. 6, Jun. 2018.
- [57]. Kim YJ et al. , “PAIP 2019: Liver cancer segmentation challenge,” Med. Image Anal, vol. 67, Jan. 2021, Art. no. 101854.
- [58]. Otsu N, “A threshold selection method from gray-level histograms,” IEEE Trans. Syst., Man, Cybern, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.
- [59]. Simonyan K and Zisserman A, “Very deep convolutional networks for large-scale image recognition,” in Proc. ICLR, San Diego, CA, USA, 2015.
- [60]. Deng J, Dong W, Socher R, Li L-J, Li K, and Fei-Fei L, “ImageNet: A large-scale hierarchical image database,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Miami, FL, USA, Jun. 2009, pp. 248–255.
- [61]. Simonyan K and Zisserman A, “Very deep convolutional networks for large-scale image recognition,” in Proc. ICLR, San Diego, CA, USA, May 2015. [Online]. Available: <http://arxiv.org/abs/1409.1556>



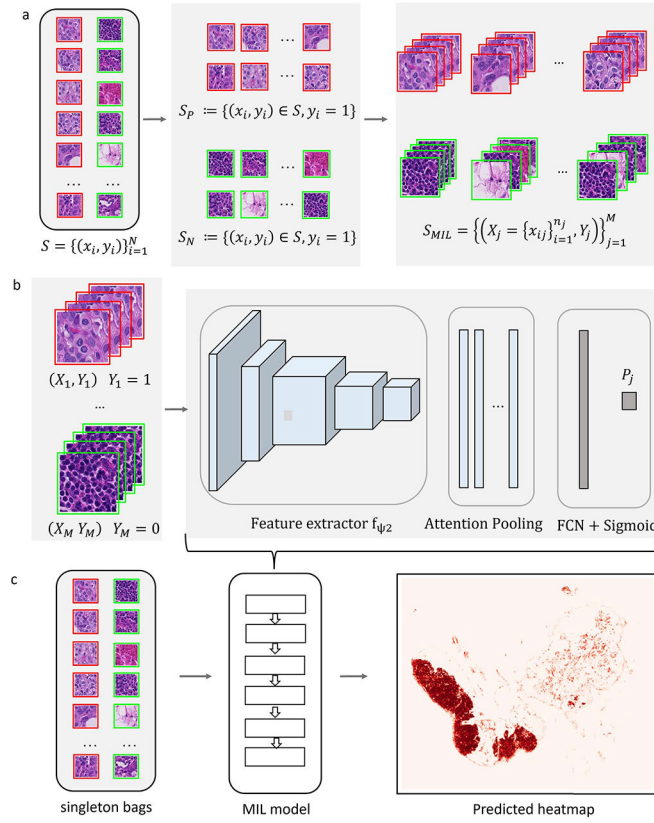
**Fig. 1.** Example of the refining coarse annotation of breast cancer metastasis made by pathologists on a WSI of lymph node section, without requiring any external training data.



**Fig. 2.** Illustration of the "noisy" label problem in coarse annotations of WSI.



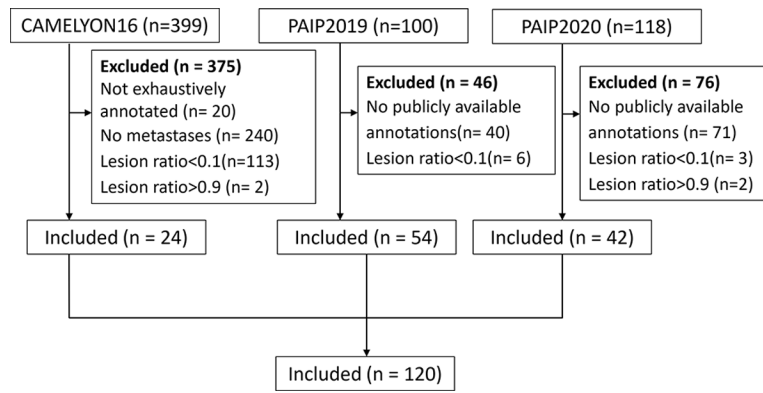
**Fig. 3.**  
WSI patching and patch-level noisy labels assignment.



**Fig. 4. Overview of the LC-MIL framework.**

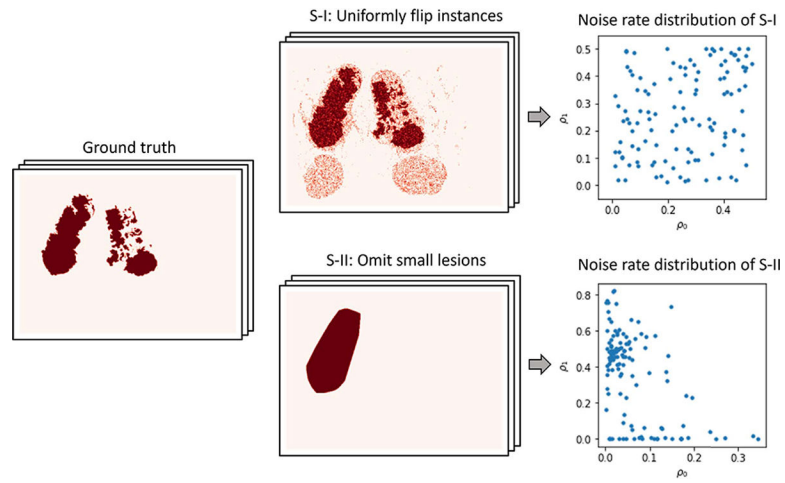
(a) MIL dataset construction: patches in WSI are split into two subsets  $S_P$  and  $S_N$  based on the original coarse annotations. (b) An attention based MIL model is trained on bags  $X_i = \{x_{ij}\}_{j=1}^{n_i}$ . (c) At inference, each patch is considered as a singleton bag and receives risk score to generate a predicted heatmap.





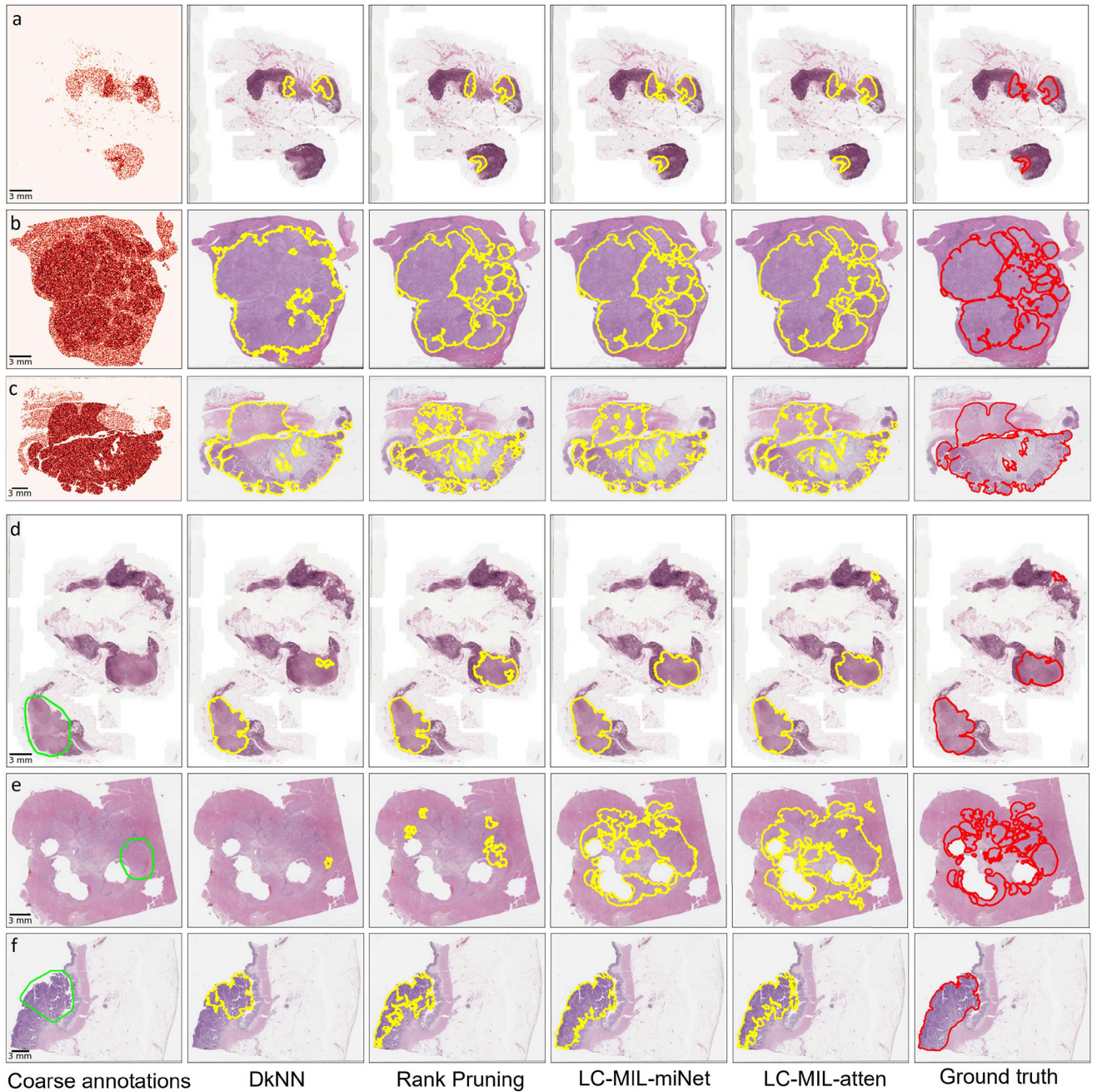
**Fig. 5. Data inclusion diagram.**

A total of 120 slides are included. From these, 24 are lymph node sections from breast cancer patients (CAMELYON16), 54 are liver cancer resection samples (PAIP2019), and the remaining 42 are colorectal cancer resection samples (PAIP2020).



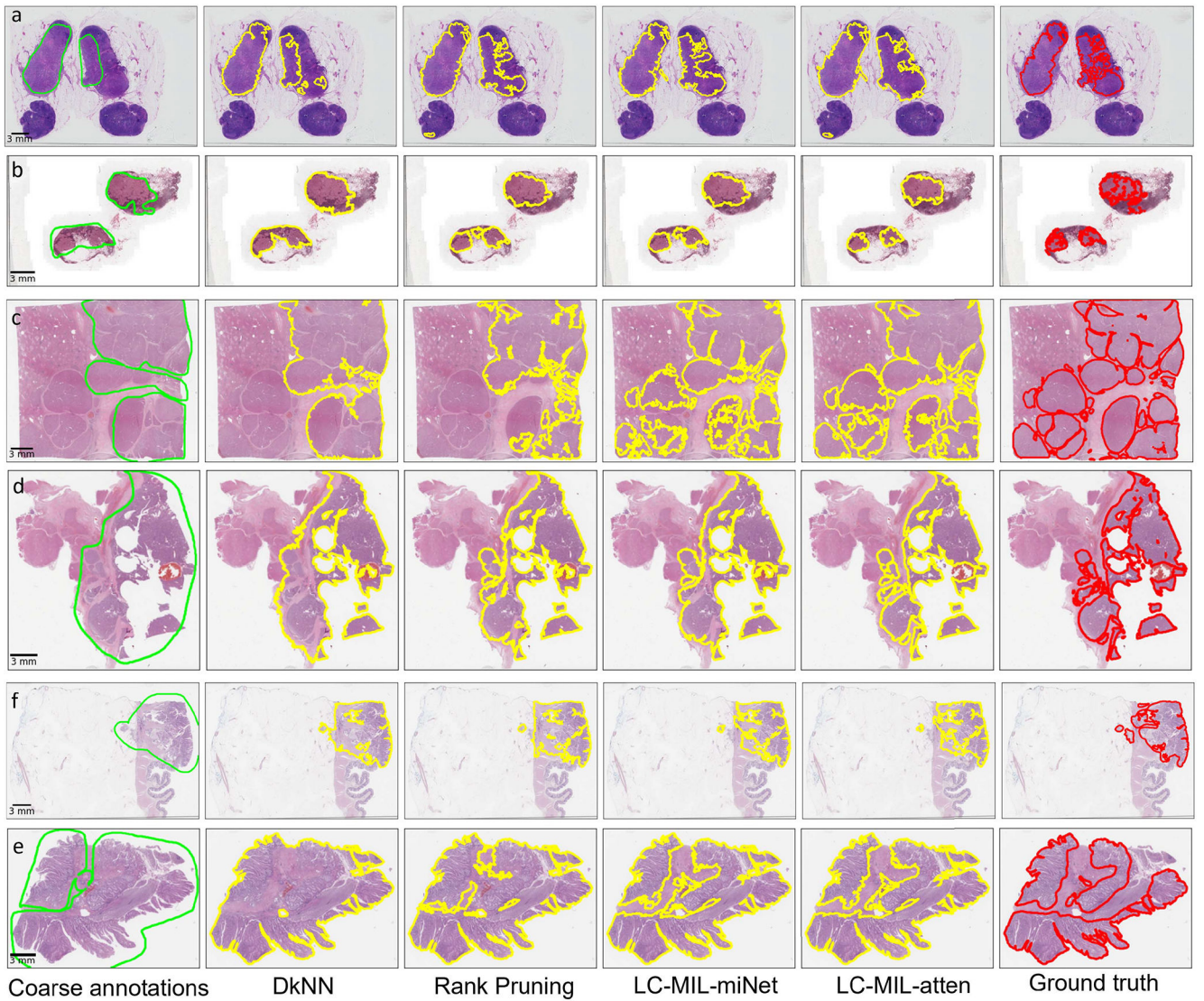
**Fig. 6. Generation of synthetic coarse annotations**

S-I: Coarse annotations generated by randomly (and uniformly) flipping patches' labels;  
 S-II: Coarse annotations generated by omitting small lesions.  $\rho_0$  represents the fraction of mislabeled instances in true negative instances; and  $\rho_1$  denotes the fraction of mislabeled instances in true positive instances.



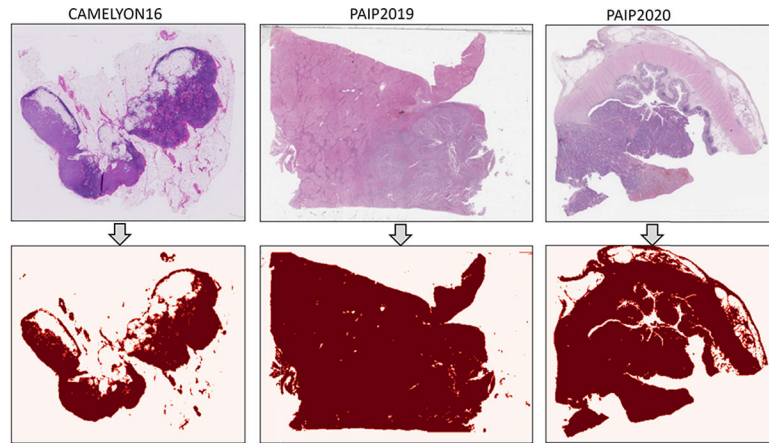
**Fig. 7. Examples of synthetic coarse annotations and refinement.**

(a-c) S-I: synthetic coarse annotations generated by uniformly flipping patches; (d-f): S-II: synthetic coarse annotations generated by omitting small lesions. Among those examples, (a) and (d) are from CAMELYON16, (b) and (e) are from PAIP2019, (c) and (f) are from PAIP2020. From left to right, six versions (coarse annotations (1st column, heatmap/lime lines); predicted contours using DkNN (2nd column, yellow lines); predicted contours using Rank Pruning (3rd column, yellow lines); predicted contours using LC-MIL-miNet (4th column, yellow lines)); predicted contours using LC-MIL-atten (5th column, yellow lines)); ground truth (6th column, red lines)) of cancerous region contours are shown.



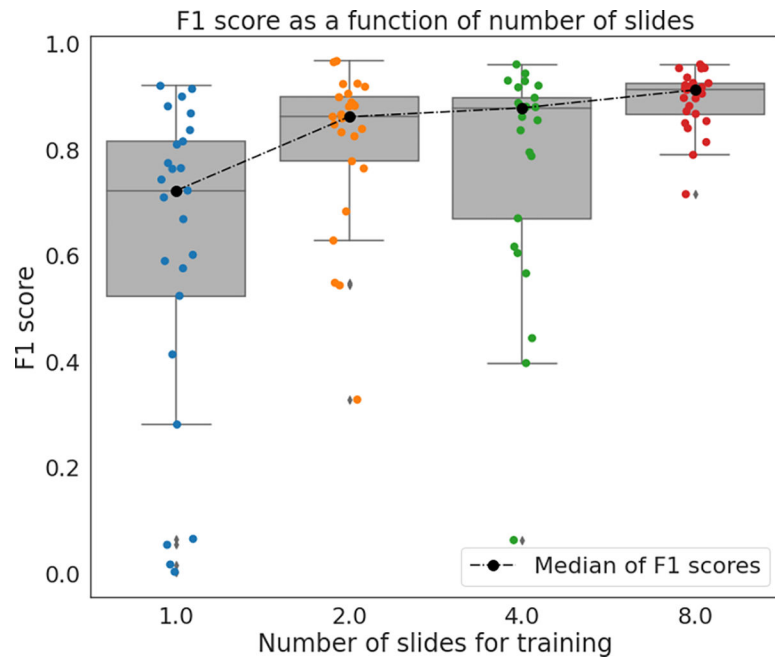
**Fig. 8. Examples of hand-drawn coarse annotations and refinement.**

Among those examples, (a) and (b) are from CAMELYON16, (c) and (d) are from PAIP2019, (e) and (f) are from PAIP2020. From left to right, six versions (coarse annotations (1st column, lime lines); predicted contours using DkNN (2nd column, yellow lines); predicted contours using Rank Pruning (3rd column, yellow lines); predicted contours using LC-MIL-miNet (4th column, yellow lines)); predicted contours using LC-MIL-atten (5th column, yellow lines); ground truth (6th column, red lines)) of cancerous regions contours are shown.

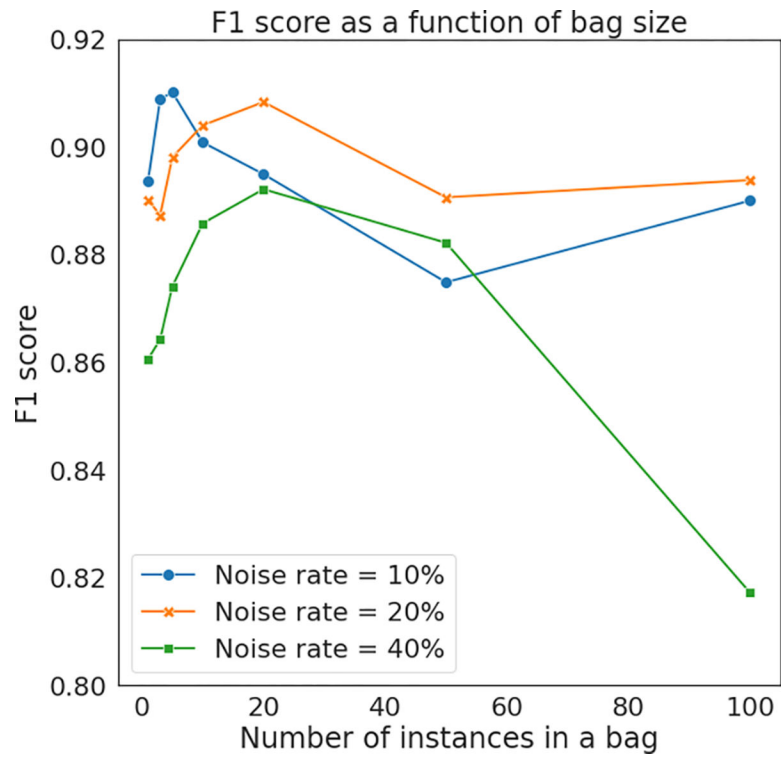


**Fig. 9. Examples of tissue region identification.**

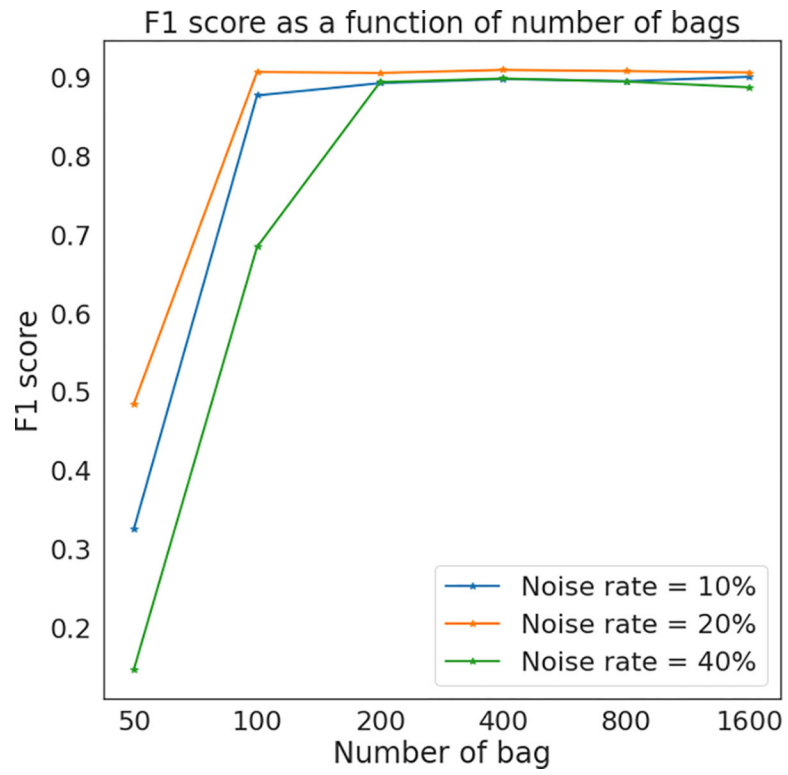
One example of each subset is shown. The first rows shows thumbnails of WSIs, and the second row shows the corresponding tissue masks.



**Fig. 10.** Prediction results using multiple slides for training.

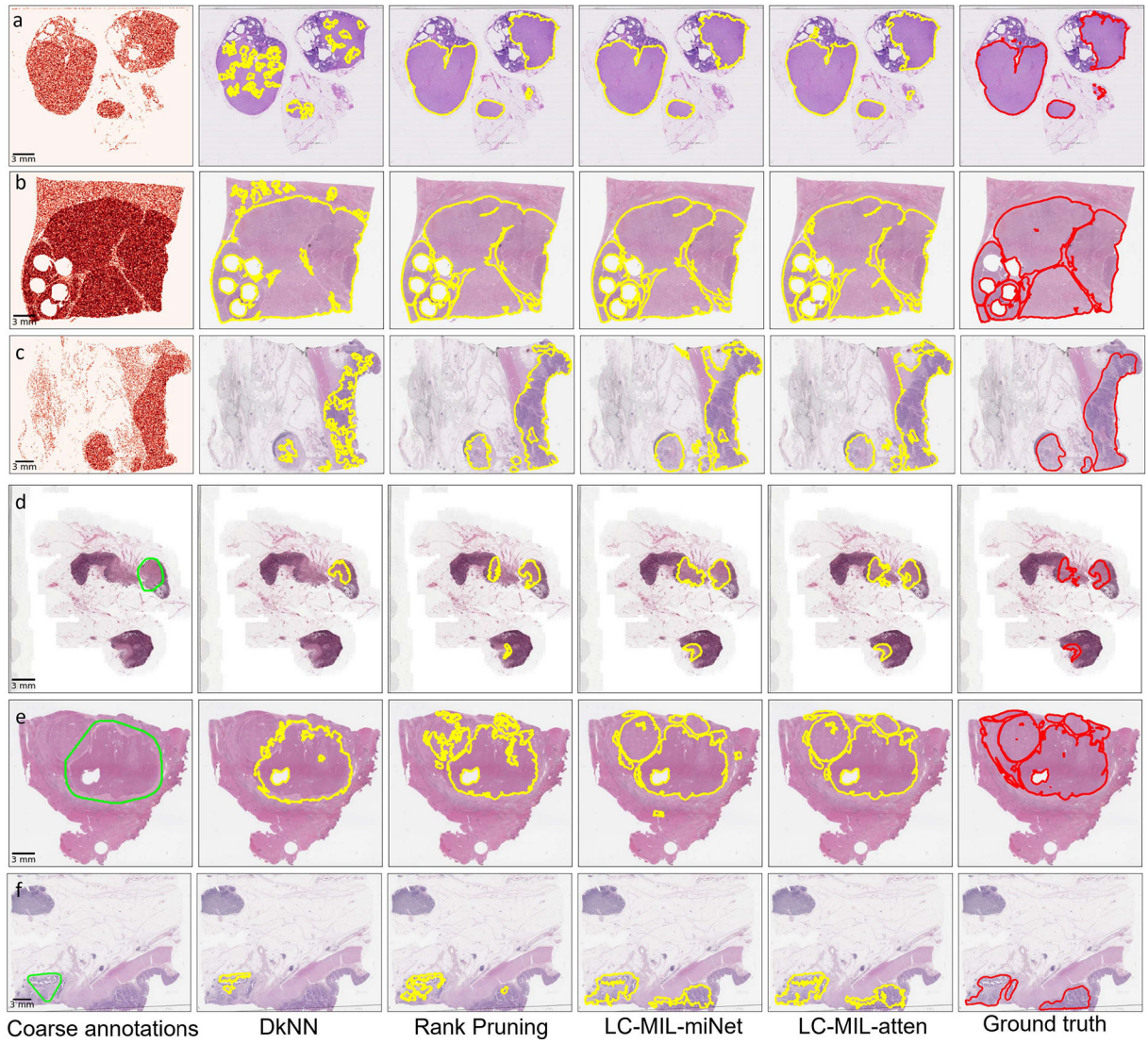


**Fig. 11.** Prediction results using various bag size.



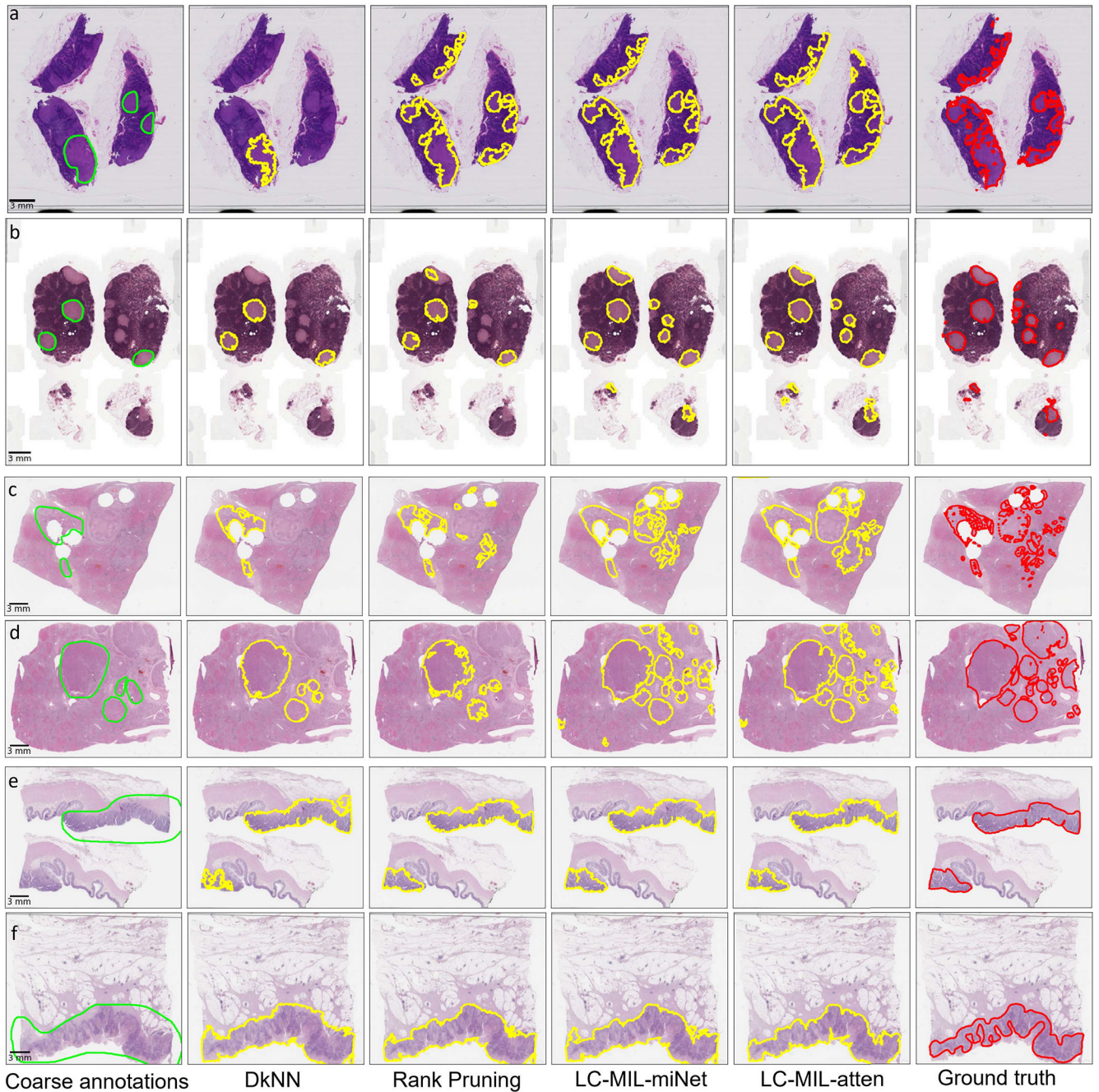
**Fig. 12.** Prediction results using various number of bags.





**Fig. 13. Supplementary examples of simulated coarse annotations and refinement.**

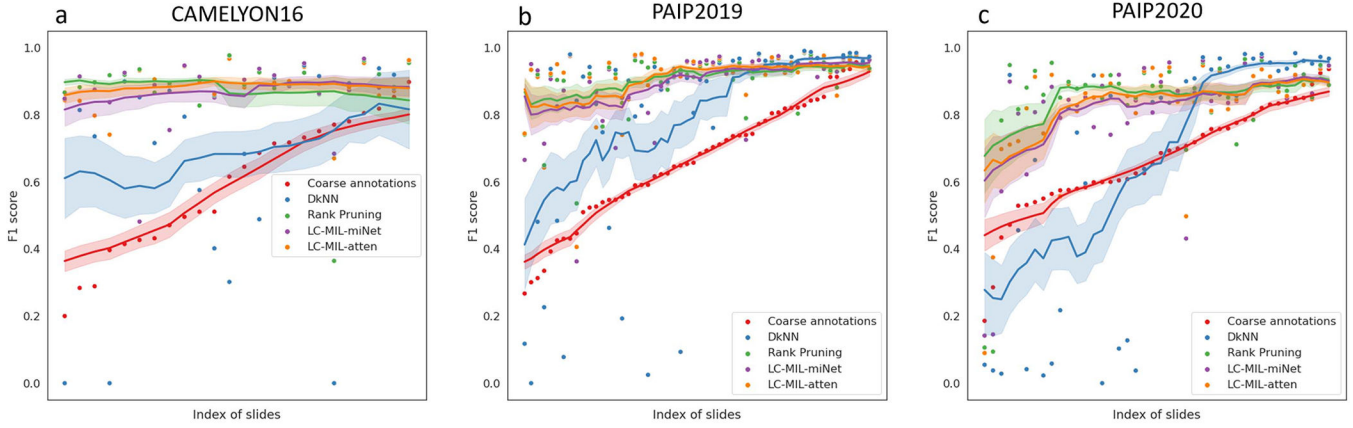
(a-c) S-I: synthetic coarse annotations generated by uniformly flipping patches; (d-f): S-II: synthetic coarse annotations generated by omitting small lesions. Among those examples, (a) and (d) are from CAMELYON16, (b) and (e) are from PAIP2019, (c) and (f) are from PAIP2020. From left to right, six versions (coarse annotations (1st column, lime lines); predicted contours using DkNN (2nd column, yellow lines); predicted contours using Rank Pruning (3rd column, yellow lines); predicted contours using LC-MIL-miNet (4th column, yellow lines); predicted contours using LC-MIL-atten (5th column, yellow lines)); ground truth (6th column, red lines)) of cancerous regions contours are shown.



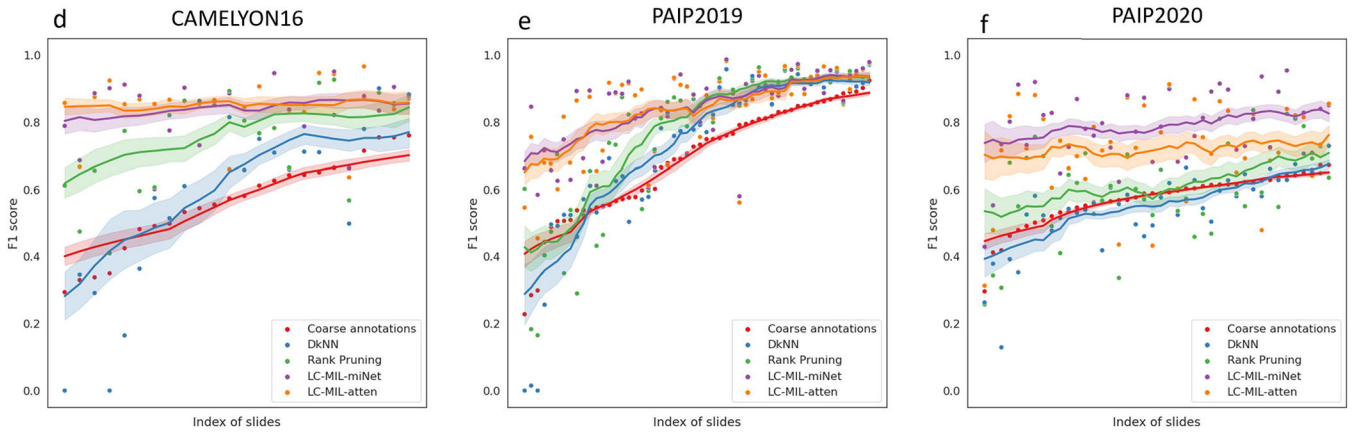
**Fig. 14. Supplementary examples of hand-drawn coarse annotations and refinement.**

Among those examples, (a) and (b) are from CAMELYON16, (c) and (d) are from PAIP2019, (e) and (f) are from PAIP2020. From left to right, six versions (coarse annotations (1st column, lime lines); predicted contours using DkNN (2nd column, yellow lines); predicted contours using Rank Pruning (3rd column, yellow lines); predicted contours using LC-MIL-miNet (4th column, yellow lines)); predicted contours using LC-MIL-atten (5th column, yellow lines)); ground truth (6th column, red lines)) of cancerous regions contours are shown.

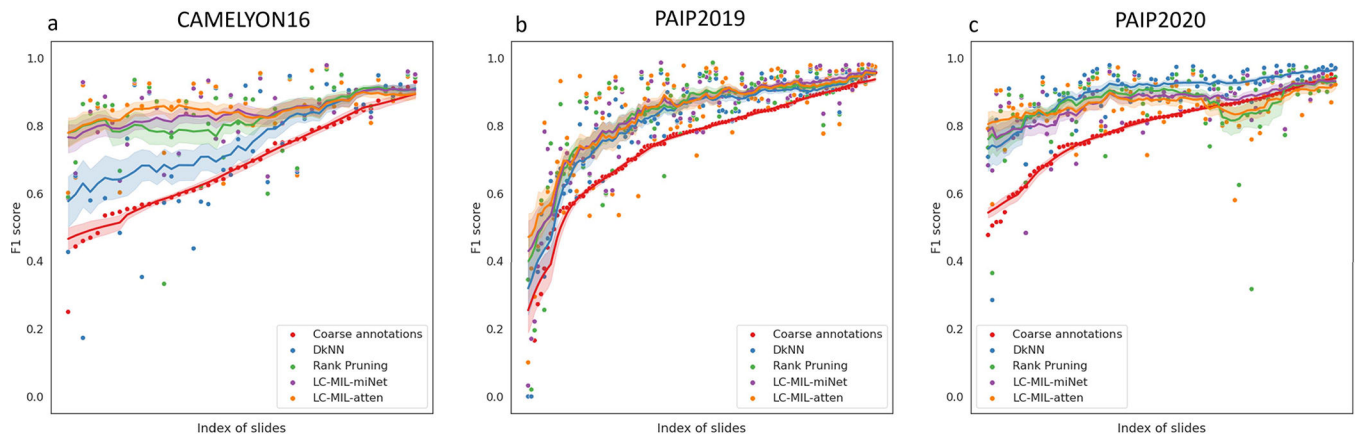
S-I: Uniformly Flip Instances



S-II: Omit Small Lesions



**Fig. 15. Refinement performance under different noise conditions (synthetic annotations).** F1 scores of coarse annotations (red), refined annotations by DkNN (blue), Rank Pruning (green), LC-MIL-miNet (purple), and LC-MIL-atten (orange), with slides sorted by F1 scores of their coarse annotations. The moving average (solid lines), where window size  $k=15$ , and the 68% confidence interval for  $k$  observations (shaded areas) are also shown.



**Fig. 16. Refinement performance under different noise conditions (hand-drawn annotations).** F1 scores of coarse annotations and refined annotations (using different methods), with slides are sorted by F1 scores of their coarse annotations. The moving average (solid lines), where window size  $k = 15$ , and the 68% confidence interval for  $k$  observations (shaded areas) are also shown.

TABLE I

Summary of Refinement on Synthetic Coarse Annotations

Datasets	Methods	S-I: Uniformly Flip Instances			S-II: Omit Small Lesions		
		PPV	TPR	F1	PPV	TPR	F1
CAMELYON16	Coarse annotations	0.545 ± 0.246	0.733 ± 0.153	0.592 ± 0.195	0.573 ± 0.123	0.606 ± 0.228	0.561 ± 0.136
	DkNN	0.798 ± 0.308	0.649 ± 0.343	0.686 ± 0.313	0.810 ± 0.259	0.523 ± 0.289	0.597 ± 0.266
	Rank Pruning	0.915 ± 0.050	0.865 ± 0.148	0.879 ± 0.113	0.906 ± 0.083	0.684 ± 0.204	0.758 ± 0.144
	LC-MIL	0.836 ± 0.151	0.934 ± 0.062	0.871 ± 0.104	0.828 ± 0.120	0.880 ± 0.132	0.840 ± 0.091
	LC-MIL-atten	0.863 ± 0.110	0.915 ± 0.077	<b>0.882 ± 0.068</b>	0.833 ± 0.128	0.895 ± 0.110	<b>0.849 ± 0.084</b>
PAIP2019	Coarse annotations	0.641 ± 0.237	0.747 ± 0.151	0.665 ± 0.182	0.674 ± 0.123	0.757 ± 0.266	0.687 ± 0.168
	DkNN	0.916 ± 0.149	0.767 ± 0.317	0.790 ± 0.291	0.788 ± 0.185	0.728 ± 0.303	0.722 ± 0.242
	Rank Pruning	0.911 ± 0.114	0.915 ± 0.070	0.907 ± 0.083	0.858 ± 0.129	0.755 ± 0.275	0.768 ± 0.215
	LC-MIL	0.875 ± 0.156	0.942 ± 0.058	0.875 ± 0.156	0.828 ± 0.151	0.889 ± 0.137	<b>0.842 ± 0.119</b>
	LC-MIL-atten	0.895 ± 0.138	0.939 ± 0.066	<b>0.907 ± 0.098</b>	0.884 ± 0.129	0.826 ± 0.177	0.834 ± 0.128
PAIP2020	Coarse annotations	0.545 ± 0.246	0.697 ± 0.153	0.592 ± 0.195	0.682 ± 0.098	0.497 ± 0.079	0.570 ± 0.076
	DkNN	0.901 ± 0.220	0.626 ± 0.387	0.662 ± 0.372	0.882 ± 0.086	0.408 ± 0.113	0.548 ± 0.118
	Rank Pruning	0.900 ± 0.197	0.807 ± 0.165	<b>0.841 ± 0.176</b>	0.910 ± 0.092	0.477 ± 0.173	0.607 ± 0.147
	LC-MIL	0.820 ± 0.198	0.836 ± 0.202	0.811 ± 0.181	0.874 ± 0.092	0.736 ± 0.163	<b>0.786 ± 0.106</b>
	LC-MIL-atten	0.833 ± 0.191	0.848 ± 0.175	0.823 ± 0.167	0.881 ± 0.108	0.638 ± 0.192	0.718 ± 0.138

**TABLE II**

Run-Time Comparison (min)

Methods	CAMELYON16	PAIP2019	PAIP2020
DkNN	$6.94 \pm 3.07$	$38.9 \pm 18.3$	$15.8 \pm 3.60$
Rank Pruning	$11.3 \pm 2.69$	$34.9 \pm 18.1$	$17.8 \pm 3.90$
LC-MIL-miNet	$5.60 \pm 1.52$	$18.2 \pm 5.95$	$9.84 \pm 1.65$
LC-MIL-atten	$5.70 \pm 1.48$	$18.8 \pm 5.87$	$11.8 \pm 1.94$

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

TABLE III

Summary of Refinement on Hand-Drawn Coarse Annotations

Dataset	Method	PPV	TPR	F1
CAMELYON16	Coarse annotations	0.656 ± 0.219	0.830 ± 0.208	0.682 ± 0.149
	DKNN	0.832 ± 0.167	0.776 ± 0.252	0.756 ± 0.182
	Rank Pruning	0.868 ± 0.131	0.836 ± 0.168	0.830 ± 0.121
	LC-MIL	0.819 ± 0.141	0.898 ± 0.129	0.840 ± 0.096
	LC-MIL-miNet LC-MIL-atten	0.828 ± 0.140	0.901 ± 0.114	<b>0.848 ± 0.096</b>
PAIP2019	Coarse annotations	0.647 ± 0.202	0.934 ± 0.174	0.742 ± 0.182
	DKNN	0.753 ± 0.202	0.920 ± 0.179	0.808 ± 0.181
	Rank Pruning	0.799 ± 0.194	0.913 ± 0.155	0.830 ± 0.168
	LC-MIL	0.785 ± 0.202	0.932 ± 0.131	<b>0.833 ± 0.167</b>
	LC-MIL-miNet LC-MIL-atten	0.805 ± 0.213	0.919 ± 0.114	<b>0.833 ± 0.163</b>
PAIP2020	Coarse annotations	0.705 ± 0.152	0.951 ± 0.092	0.796 ± 0.114
	DKNN	0.882 ± 0.105	0.936 ± 0.112	<b>0.898 ± 0.095</b>
	Rank Pruning	0.908 ± 0.091	0.853 ± 0.152	0.867 ± 0.110
	LC-MIL	0.860 ± 0.112	0.897 ± 0.103	0.869 ± 0.081
	LC-MIL-miNet LC-MIL-atten	0.902 ± 0.097	0.844 ± 0.114	0.863 ± 0.077

TABLE IV

Summary of Refinement on S-I: Uniformly Flip Patches

Dataset	Tumor/tissue	Method	F1	IoU	PPV	TPR	TNR	NPV
CAMELYON16	29.7 % ( $\pm$ 17.0 %)	Coarse annotations	0.592 $\pm$ 0.195	0.416 $\pm$ 0.189	0.545 $\pm$ 0.246	0.733 $\pm$ 0.153	0.745 $\pm$ 0.137	0.869 $\pm$ 0.139
		DkNN	0.686 $\pm$ 0.313	0.570 $\pm$ 0.297	0.798 $\pm$ 0.308	0.649 $\pm$ 0.343	0.971 $\pm$ 0.038	0.889 $\pm$ 0.146
		Rank Pruning	0.879 $\pm$ 0.113	0.772 $\pm$ 0.147	0.915 $\pm$ 0.050	0.865 $\pm$ 0.148	0.965 $\pm$ 0.033	0.956 $\pm$ 0.040
		LC-MIL	0.871 $\pm$ 0.104	0.758 $\pm$ 0.144	0.836 $\pm$ 0.151	0.934 $\pm$ 0.062	0.912 $\pm$ 0.147	0.971 $\pm$ 0.037
		LC-MIL-mlNet	<b>0.882 <math>\pm</math> 0.068</b>	0.770 $\pm$ 0.117	0.863 $\pm$ 0.110	0.915 $\pm$ 0.077	0.938 $\pm$ 0.059	0.967 $\pm$ 0.042
PAIP2019	38.0 % ( $\pm$ 18.9 %)	Coarse annotations	0.665 $\pm$ 0.182	0.514 $\pm$ 0.197	0.641 $\pm$ 0.237	0.747 $\pm$ 0.151	0.758 $\pm$ 0.157	0.819 $\pm$ 0.144
		DkNN	0.790 $\pm$ 0.291	0.711 $\pm$ 0.297	0.916 $\pm$ 0.149	0.767 $\pm$ 0.317	0.961 $\pm$ 0.057	0.905 $\pm$ 0.140
		Rank Pruning	0.907 $\pm$ 0.083	0.823 $\pm$ 0.121	0.911 $\pm$ 0.114	0.915 $\pm$ 0.070	0.952 $\pm$ 0.056	0.950 $\pm$ 0.039
		LC-MIL	0.875 $\pm$ 0.156	0.811 $\pm$ 0.144	0.875 $\pm$ 0.156	0.942 $\pm$ 0.058	0.919 $\pm$ 0.164	0.954 $\pm$ 0.054
		LC-MIL-mlNet	<b>0.907 <math>\pm</math> 0.098</b>	0.828 $\pm$ 0.133	0.895 $\pm$ 0.138	0.939 $\pm$ 0.066	0.934 $\pm$ 0.130	0.957 $\pm$ 0.046
PAIP2020	43.3 % ( $\pm$ 13.0 %)	Coarse annotations	0.592 $\pm$ 0.195	0.503 $\pm$ 0.169	0.545 $\pm$ 0.246	0.697 $\pm$ 0.153	0.750 $\pm$ 0.138	0.765 $\pm$ 0.157
		DkNN	0.662 $\pm$ 0.372	0.588 $\pm$ 0.362	0.901 $\pm$ 0.220	0.626 $\pm$ 0.387	0.965 $\pm$ 0.058	0.830 $\pm$ 0.168
		Rank Pruning	<b>0.841 <math>\pm</math> 0.176</b>	0.744 $\pm$ 0.182	0.900 $\pm$ 0.197	0.807 $\pm$ 0.165	0.933 $\pm$ 0.130	0.867 $\pm$ 0.116
		LC-MIL	0.811 $\pm$ 0.181	0.704 $\pm$ 0.195	0.820 $\pm$ 0.198	0.836 $\pm$ 0.202	0.881 $\pm$ 0.120	0.882 $\pm$ 0.124
		LC-MIL-mlNet	0.823 $\pm$ 0.167	0.718 $\pm$ 0.185	0.833 $\pm$ 0.191	0.848 $\pm$ 0.175	0.886 $\pm$ 0.128	0.882 $\pm$ 0.131



TABLE V

Summary of Refinement on S-II: Omit Small Lesions

Dataset	Tumor/tissue	Method	F1	IoU	PPV	TPR	TNR	NPV
CAMELYON16	29.7% ( $\pm 17.0\%$ )	Coarse annotations	0.561 $\pm$ 0.136	0.402 $\pm$ 0.129	0.485 $\pm$ 0.147	0.927 $\pm$ 0.177	0.987 $\pm$ 0.012	0.984 $\pm$ 0.020
		DKNN	0.597 $\pm$ 0.266	0.471 $\pm$ 0.247	0.774 $\pm$ 0.188	0.861 $\pm$ 0.233	0.998 $\pm$ 0.004	0.981 $\pm$ 0.023
		Rank Pruning	0.758 $\pm$ 0.144	0.630 $\pm$ 0.175	0.844 $\pm$ 0.147	0.869 $\pm$ 0.174	0.998 $\pm$ 0.003	0.986 $\pm$ 0.019
		LC-MIL	0.840 $\pm$ 0.091	0.735 $\pm$ 0.127	0.773 $\pm$ 0.142	0.922 $\pm$ 0.123	0.996 $\pm$ 0.005	0.993 $\pm$ 0.014
		LC-MIL-atten	<b>0.849 <math>\pm</math> 0.084</b>	0.747 $\pm$ 0.117	0.827 $\pm$ 0.137	0.906 $\pm$ 0.119	0.995 $\pm$ 0.008	0.994 $\pm$ 0.012
PAIP2019	38.0% ( $\pm 18.9\%$ )	Coarse annotations	0.687 $\pm$ 0.168	0.547 $\pm$ 0.186	0.586 $\pm$ 0.179	0.989 $\pm$ 0.029	0.892 $\pm$ 0.076	0.930 $\pm$ 0.093
		DKNN	0.722 $\pm$ 0.242	0.612 $\pm$ 0.254	0.713 $\pm$ 0.193	0.971 $\pm$ 0.051	0.947 $\pm$ 0.051	0.929 $\pm$ 0.093
		Rank Pruning	0.768 $\pm$ 0.215	0.666 $\pm$ 0.245	0.766 $\pm$ 0.195	0.955 $\pm$ 0.055	0.962 $\pm$ 0.043	0.931 $\pm$ 0.094
		LC-MIL	<b>0.842 <math>\pm</math> 0.119</b>	0.743 $\pm$ 0.165	0.747 $\pm$ 0.208	0.961 $\pm$ 0.053	0.955 $\pm$ 0.040	0.956 $\pm$ 0.069
		LC-MIL-atten	0.834 $\pm$ 0.128	0.734 $\pm$ 0.173	0.781 $\pm$ 0.219	0.934 $\pm$ 0.077	0.951 $\pm$ 0.049	0.955 $\pm$ 0.067
PAIP2020	43.3% ( $\pm 13.0\%$ )	Coarse annotations	0.570 $\pm$ 0.076	0.403 $\pm$ 0.070	0.600 $\pm$ 0.130	0.979 $\pm$ 0.045	0.961 $\pm$ 0.030	0.918 $\pm$ 0.058
		DKNN	0.548 $\pm$ 0.118	0.386 $\pm$ 0.104	0.834 $\pm$ 0.116	0.968 $\pm$ 0.048	0.991 $\pm$ 0.010	0.909 $\pm$ 0.058
		Rank Pruning	0.607 $\pm$ 0.147	0.452 $\pm$ 0.152	0.867 $\pm$ 0.101	0.901 $\pm$ 0.129	0.993 $\pm$ 0.009	0.916 $\pm$ 0.062
		LC-MIL	<b>0.786 <math>\pm</math> 0.106</b>	0.659 $\pm$ 0.138	0.832 $\pm$ 0.115	0.923 $\pm$ 0.057	0.985 $\pm$ 0.014	0.950 $\pm$ 0.055
		LC-MIL-atten	0.718 $\pm$ 0.138	0.576 $\pm$ 0.157	0.884 $\pm$ 0.100	0.875 $\pm$ 0.075	0.988 $\pm$ 0.014	0.933 $\pm$ 0.064

TABLE VI

Summary of Refinement on Hand-Drawn Annotations

Dataset	Tumor/tissue	Method	F1	IoU	PPV	TPR	TNR	NPV
CAME LYON16	29.7 % ( $\pm$ 17.0 %)	Coarse annotations	0.682 $\pm$ 0.149	0.537 $\pm$ 0.173	0.656 $\pm$ 0.219	0.830 $\pm$ 0.208	0.980 $\pm$ 0.034	0.992 $\pm$ 0.015
		DKNN	0.756 $\pm$ 0.182	0.639 $\pm$ 0.214	0.832 $\pm$ 0.167	0.776 $\pm$ 0.252	0.992 $\pm$ 0.020	0.990 $\pm$ 0.017
		Rank Pruning	0.830 $\pm$ 0.121	0.725 $\pm$ 0.154	0.868 $\pm$ 0.131	0.836 $\pm$ 0.168	0.995 $\pm$ 0.009	0.993 $\pm$ 0.013
		LC-MIL	0.840 $\pm$ 0.096	0.736 $\pm$ 0.136	0.819 $\pm$ 0.141	0.898 $\pm$ 0.129	0.994 $\pm$ 0.010	0.994 $\pm$ 0.014
		LC-MIL-miNet	<b>0.848 <math>\pm</math> 0.096</b>	0.748 $\pm$ 0.133	0.828 $\pm$ 0.140	0.901 $\pm$ 0.114	0.994 $\pm$ 0.012	0.995 $\pm$ 0.012
PAIP2019	38.0 % ( $\pm$ 18.9 %)	Coarse annotations	0.742 $\pm$ 0.182	0.618 $\pm$ 0.195	0.647 $\pm$ 0.202	0.934 $\pm$ 0.174	0.848 $\pm$ 0.108	0.982 $\pm$ 0.039
		DKNN	0.808 $\pm$ 0.181	0.708 $\pm$ 0.203	0.753 $\pm$ 0.202	0.920 $\pm$ 0.179	0.912 $\pm$ 0.082	0.979 $\pm$ 0.040
		Rank Pruning	0.830 $\pm$ 0.168	0.737 $\pm$ 0.197	0.799 $\pm$ 0.194	0.913 $\pm$ 0.155	0.926 $\pm$ 0.086	0.975 $\pm$ 0.044
		LC-MIL	<b>0.833 <math>\pm</math> 0.167</b>	0.741 $\pm$ 0.197	0.785 $\pm$ 0.202	0.932 $\pm$ 0.131	0.933 $\pm$ 0.070	0.978 $\pm$ 0.030
		LC-MIL-miNet	<b>0.833 <math>\pm</math> 0.163</b>	0.742 $\pm$ 0.203	0.805 $\pm$ 0.213	0.919 $\pm$ 0.114	0.938 $\pm$ 0.075	0.971 $\pm$ 0.034
PAIP2020	43.3 % ( $\pm$ 13.0 %)	Coarse annotations	0.796 $\pm$ 0.114	0.675 $\pm$ 0.147	0.705 $\pm$ 0.152	0.951 $\pm$ 0.092	0.925 $\pm$ 0.075	0.990 $\pm$ 0.022
		DKNN	<b>0.898 <math>\pm</math> 0.095</b>	0.826 $\pm$ 0.126	0.882 $\pm$ 0.105	0.936 $\pm$ 0.112	0.977 $\pm$ 0.034	0.990 $\pm$ 0.022
		Rank Pruning	0.867 $\pm$ 0.110	0.779 $\pm$ 0.141	0.908 $\pm$ 0.091	0.853 $\pm$ 0.152	0.984 $\pm$ 0.027	0.978 $\pm$ 0.028
		LC-MIL	0.869 $\pm$ 0.081	0.776 $\pm$ 0.115	0.860 $\pm$ 0.112	0.897 $\pm$ 0.103	0.979 $\pm$ 0.020	0.981 $\pm$ 0.027
		LC-MIL-miNet	0.863 $\pm$ 0.077	0.766 $\pm$ 0.111	0.902 $\pm$ 0.097	0.844 $\pm$ 0.114	0.986 $\pm$ 0.019	0.971 $\pm$ 0.035
		LC-MIL-atten						