Multiple Instance Learning via Iterative Self-Paced Supervised Contrastive Learning

Kangning Liu [*1] **Weicheng Zhu** [*1] **Yiqiu Shen** [1] **Sheng Liu** [1] **Narges Razavian** [2]
Krzysztof J. Geras [†1 2] **Carlos Fernandez-Granda** [†1 3]

Abstract

Existing multiple instance learning (MIL) models typically consist of a feature extractor that computes a representation for each instance and an aggregator that combines instance-level features to make a bag-level prediction. In applications such as medical diagnosis, end-to-end training is often intractable due to memory constraints. It is therefore essential to divide the learning into pretraining the feature extractor and training the aggregator. Recent works have shown promising results using contrastive self-supervised learning, which operates by pushing apart representations corresponding to randomly-selected instances. For MIL binary-classification tasks, a setting of critical importance in medicine, this is problematic as randomly-selected instances mostly belong to the same class, which precludes the method from learning inter-class differences. To address this issue, we propose a novel framework, Iterative Self-paced Supervised Contrastive Learning for MIL Representations (*ItS2CLR*), which alternates between (1) training the aggregator on instance-level features, (2) estimating instance-level pseudo labels, and (3) using these pseudo labels to finetune the feature extractor. The framework employs a novel self-paced sampling strategy to ensure reliability of the pseudo labels. We evaluate *ItS2CLR* on three real-world medical datasets, showing that it improves the quality of both pseudo labels and instance-level features, and outperforms existing MIL methods in both bag and instance level accuracy.

[*]Equal contribution [†]Joint last author [1]Center for Data Science, New York University [2]New York University School of Medicine [3]Courant Institute of Mathematical Sciences, New York University. Correspondence to: Kangning Liu <kl3141@nyu.edu>, Weicheng Zhu <jackzhu@nyu.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

1. Introduction

Multiple instance learning (MIL) is a type of supervised learning where the data are arranged in bags of instances. In MIL binary-classification tasks, each instance is either positive or negative, but these instance-level labels are not available during training. Only bag-level labels are available: a bag is labeled as positive if it contains *any* positive instances, and negative otherwise. An important application of MIL is automatic cancer diagnosis from histopathology slides. Each slide is divided into hundreds or thousands of tiles but typically only slide-level labels are available (Courtiol et al., 2018; Campanella et al., 2019; Li et al., 2021; Chen & Krishnan, 2022; Zhang et al., 2022; Lu et al., 2021).

In applications with high memory usage, such as histopathology slides, end-to-end training of deep neural networks is often infeasible. Consequently, state-of-the-art approaches (Campanella et al., 2019; Li et al., 2021; Zhang et al., 2022; Lu et al., 2021; Shao et al., 2021) utilize a two-stage learning pipeline: feature extraction that maps each instance to a feature vector, and aggregation that combines the feature vectors from all instances in a bag to produce a bag-level prediction (Figure 1). Interestingly, our results indicate that in cases where end-to-end training is possible, this pipeline still provides superior performance (see Section 4.3).

In this work, we focus on a fundamental challenge in MIL: how to train the feature extractor. The three main existing approaches have significant shortcomings. (1) Pretraining on large natural image datasets such as ImageNet (Shao et al., 2021; Lu et al., 2021) is problematic for medical applications as features learned from natural images may generalize poorly to other domains (Lu et al., 2020). (2) Supervised training using bag-level labels as instance-level labels can be effective if positive bags contain mostly positive instances (Lerousseau et al., 2020; Xu et al., 2019; Chikontwe et al., 2020), but in many medical datasets this is not the case (Bejnordi et al., 2017; Li et al., 2021). (3) Contrastive self-supervised learning (CSSL) outperforms prior methods (Li et al., 2021; Ciga et al., 2022), but its effectiveness is significantly compromised under class-imbalanced settings. CSSL operates by *pushing apart* the feature vectors

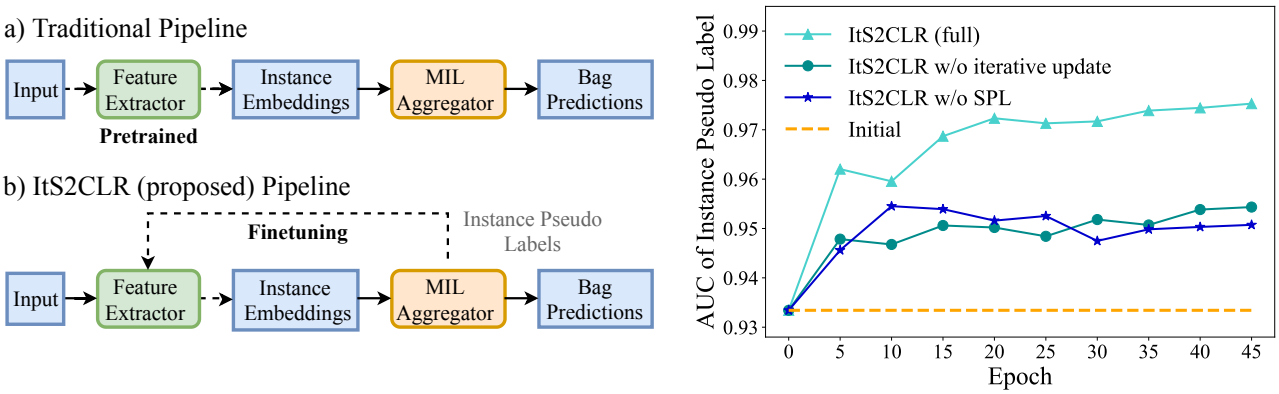

Figure 1: **Left:** (a) Traditional MIL models first pretrain a feature extractor and then train an aggregator that maps the features to a bag-level prediction. (b) Our proposed framework ItS2CLR uses instance-level pseudo labels obtained from the aggregator to finetune the feature extractor. **Right:** ItS2CLR updates the features iteratively based on a subset of the pseudo label that is selected according to a self-paced learning (SPL) strategy. On a benchmark dataset (Camelyon16 (Bejnordi et al., 2017)), this gradually improves the accuracy of the pseudo labels in terms of instance-level AUC. Both the iterative updates and SPL are important to achieve this.

of randomly selected instances. As we explain in Section 2 (see top of Figure 2), when positive bags contain mostly negative instances, CSSL training ends up mostly pushing apart negative instances, which precludes it from learning discriminative features.

Our goal is to address the shortcomings of current feature-extraction methods. We build upon several key insights. First, it is possible to extract instance-level pseudo labels from trained MIL models, which are substantially more accurate than bag-level labels. Second, we can use the pseudo labels to finetune the feature extractor, improving the instance-level features. Third, these improved features result in improved bag-level classification. Our proposed framework, illustrated in Figure 1, is called Iterative Self-Paced Supervised Contrastive Learning for MIL Representation (ItS2CLR). After initializing the features with CSSL, we iteratively improve them via supervised contrastive learning (Khosla et al., 2020) using pseudo labels estimated by the aggregator. This feature refinement utilizes a novel self-paced sampling strategy, which selects examples to ensure reliability of the pseudo labels (see Section 3.2). In summary, our contributions are the following:

1. We propose ItS2CLR – a novel MIL framework where instance features are iteratively improved using pseudo labels extracted from the MIL aggregator.
2. In order to refine the instance-level features using the pseudo labels, we utilize a novel self-paced supervised contrastive learning scheme.
3. We demonstrate that the proposed approach outperforms existing MIL methods, including end-to-end schemes, in terms of bag- and instance-level accuracy on three real-world medical datasets relevant to cancer diagnosis: two histopathology datasets and a breast ultrasound dataset.
4. We perform experiments showing that ItS2CLR is compatible with a range of aggregators, outperforms alternatives such as finetuning based on cross-entropy or end-to-end schemes, and generalizes across different scenarios where the fraction of positive instances per bag varies.

2. Contrastive Learning May Not Learn Discriminative Features In MIL

Recent MIL approaches use contrastive self-supervised learning (CSSL) to train the feature extractor (Li et al., 2021). In this section, we show that this has a crucial limitation in realistic MIL settings, which precludes it from learning discriminative features. CSSL aims to learn a representation where data from the same class are close, and data from different classes are far, without having access to class labels. This is achieved by minimizing the InfoNCE loss in Equation 1.

$$\mathcal{L}_{\text{CSSL}} = \mathbb{E}_{x,x^{\text{aug}},\{x_i^{\text{diff}}\}_{i=1}^n}\left[-\log \frac{\exp\left(f_\psi(x)\cdot f_\psi(x^{\text{aug}})/\tau\right)}{\exp\left(f_\psi(x)\cdot f_\psi(x^{\text{aug}})/\tau\right)+\sum_{i=1}^n \exp\left(f_\psi(x)\cdot f_\psi(x_i^{\text{diff}})/\tau\right)}\right], \tag{1}$$

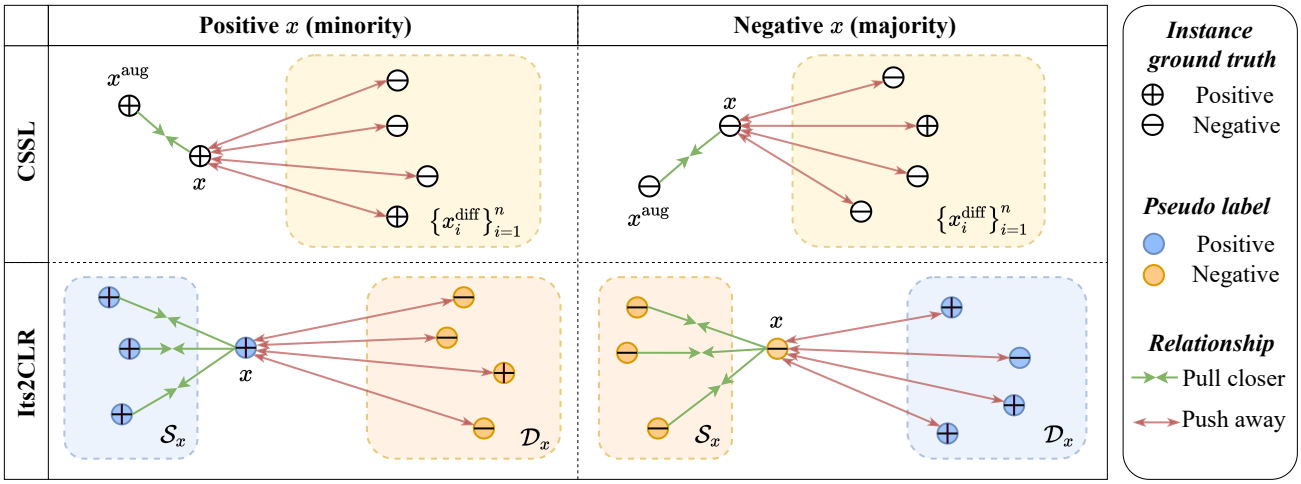

Figure 2: **Top**: In CSSL, the representation of an instance x is pulled closer to that of a random augmentation of x, x^{aug}, and pushed away from random augmentations of other randomly selected instances $\{x_i^{\text{diff}}\}_{i=1}^n$. In many MIL datasets relevant to medical diagnosis, most instances are negative, so CSSL mostly pushes apart representations of negative instances (right). **Bottom**: Our proposed framework ItS2CLR applies the supervised contrastive learning described in Section 3.1. Instance-level pseudo labels are used to build a set of positive pairs $\mathcal{S}_x$ and a set of negative pairs $\mathcal{D}_x$ corresponding to x. The representations of an instance x is pulled closer to those in $\mathcal{S}_x$ and pushed away from those in $\mathcal{D}_x$. Note that the pseudo labels can be noisy, but this noise is controlled by the self-paced sampling strategy described in Section 3.2.

where $f : \mathbb{R}^m \to \mathbb{R}^d$ is the feature extractor mapping the m-dimensional input data to a d-dimensional feature vector, $\psi : \mathbb{R}^d \to \mathbb{R}^{d'}$ is a projection head with a two-layer feed-forward network and ℓ_2 normalization, and $f_\psi = f \circ \psi$, τ is a temperature hyperparameter. The expectation is taken over the input example $x \in \mathbb{R}^m$, which is chosen at random from the training set. Minimizing the loss brings the feature vector corresponding to x closer to the feature vector corresponding to a random augmentation of x, x^{aug}, and pushes it away from the feature vectors corresponding to n other examples $\{x_i^{\text{diff}}\}_{i=1}^n$ sampled from the training set.

A key assumption in CSSL is that x belongs to a different class than most of $x_1^{\text{diff}}, \ldots, x_n^{\text{diff}}$. This is usually the case in standard classification datasets with many classes such as ImageNet (Deng et al., 2009), but *not at all in MIL tasks relevant to medical diagnosis*. In such cases, a majority of instances are negative (e.g. 95% in Camelyon16). As a result, most terms in the sum $\sum_{i=1}^n \exp\left(f_\psi(x) \cdot f_\psi(x_i^{\text{diff}})/\tau\right)$ of the loss in Equation 1 correspond to pairs of examples (x, x_i^{diff}) where both belong to the negative class. Therefore, minimizing the loss mostly pushes apart the representations of negative instances, as illustrated in the top panel of Figure 2. This is an example of *class collision* (Arora et al., 2019; Chuang et al., 2020), a general problem in CSSL, which has been shown to impair performance on downstream tasks (Ash et al., 2021; Zheng et al., 2021).

Class collision makes CSSL learn representations that are not discriminative between classes. In order to study this phenomenon, we report the average inter-class distances and intra-class deviations for features learned by CSSL on Camelyon16 in Table 1. As predicted by our analysis above, the inter-class deviations corresponding to the negative instances is very large. In fact, it is greater than the average distance between positive and negative distances! In contrast, the intra-class deviation of the features learned by our proposed framework ItS2CLR is substantially smaller than both the corresponding inter-class distance and the intra-class deviation of CSSL. This suggests that the features learned by ItS2CLR are more discriminative, which is confirmed by the results in Section 4.

In order to avoid class collision in MIL, a tempting alternative is to use the bag-level labels. Unfortunately, this is ineffective for some MIL datasets relevant to medical diagnosis, where positive bags contain just a small fraction of positive instances (10% for Camelyon16 (Bejnordi et al., 2017)). Consequently, even if we select $\{x_i^{\text{diff}}\}_{i=1}^n$ from the positive bags in equation 1 when x is negative, most of the selected instances will still be negative. Overcoming the class-collision problem requires explicitly detecting positive instances. This motivates our proposed framework, described in the following section.

3. MIL via Iterative Self-paced Supervised Contrastive learning

Iterative **S**elf-paced **S**upervised **C**ontrastive **L**earning for MIL **R**epresentations (ItS2CLR) addresses the limitation of MIL based on contrastive self-supervised learning (CSSL)

Table 1: Quantitative analysis of instance-level features learned from Camelyon16 (Bejnordi et al., 2017). The inter-class distance is the ℓ_2-norm distance between the mean feature vector of the positive instances and that of the negative instances. The intra-class deviation is the square root of the spectral norm of the covariance matrix of the features corresponding to each class. The spectral norm is the largest eigenvalue of the covariance matrix and is therefore equal to the variance in the direction of greatest variance. Due to class collision among negative instances in CSSL (see Section 2), the intra-class deviation of the corresponding features is very large, and even larger than the inter-class distance. In contrast, the features learned by the proposed framework ItS2CLR have smaller intra-class deviation among both negative and positive instances, and a larger inter-class distance.

	Training set			Test set		
	Inter-class distance	Intra-class deviation		Inter-class distance	Intra-class deviation	
		pos	*neg*		*pos*	*neg*
CSSL (SimCLR)	1.835	1.687	2.110	2.109	2.006	2.202
ItS2CLR (proposed)	**2.376**	**1.383**	**0.648**	**2.432**	**1.476**	**0.717**

Algorithm 1 It2SCLR

Require: Feature extractor f, projection head ψ;
Require: MIL aggregator g_ϕ, where ϕ is an instance classifier;
Require: Bags $\{X_b\}_{b=1}^B$, bag-level labels $\{Y_b\}_{b=1}^B$.
$f^{(0)} \leftarrow f_{\text{SSL}}$ ▷ Initialize f with CSSL-pretrained weights
for $t = 0$ to T **do**
  $h_k^b \leftarrow f^{(t)}(x_k^b)$ ▷ Extract instance embeddings
  $H_b \leftarrow \{h_k^b\}_{k=1}^{K_b}$ ▷ Group instance embedding into bags
  $g_\phi^{(t)} \leftarrow$ Train with H_b and Y_b's ▷ Train the aggregator
  $\text{AUC}_{val}^{(t)} \leftarrow$ bag-level AUC on validation set
  if $\text{AUC}_{val}^{(t)} \geq \max_{t' \leq t} \left\{ \text{AUC}_{val}^{(t')} \right\}$ **then**
    $\hat{y}_k^b \leftarrow \mathbb{1}_{\{\phi^{(t)}(h_k^b) > \eta\}}$ ▷ Update instance pseudo labels
  end if
  $f_\psi^{(t+1)} \leftarrow \text{argmin}_{f_\psi} \mathcal{L}_{\text{sup}}(f_\psi^{(t)})$ ▷ Optimize Eq.(2)
end for

described in Section 2. ItS2CLR relies on latent variables indicating whether each instance is positive or negative, which we call *instance-level pseudo labels*. To estimate these pseudo labels we use instance-level class probabilities, produced by MIL aggregators. We specifically use the aggregator from DS-MIL (Li et al., 2021) (binarizing the probabilities according to a threshold $\eta \in (0, 1)$, which is a hyper-parameter), but our framework can be deployed with any other MIL module equipped with instance-level estimation.

ItS2CLR uses pseudo labels to finetune the feature extractor (initialized using CSSL). In the spirit of the expectation-maximization algorithm, we alternate between refining the feature extractor, re-estimating the pseudo labels, and training the aggregator, as described in Algorithm 1. A key challenge is that the pseudo labels are noisy, especially at the beginning. We address this by applying a contrastive loss to perform finetuning as described in Section 3.1, and only updating reliable pseudo labels according to a novel self-paced learning scheme as explained in Section 3.2. The right panel of Figure 1 shows that iteratively updating the pseudo labels using our scheme improves their accuracy on Camelyon16 (Bejnordi et al., 2017) (where ground-truth instance labels are available). In Section 4 we demonstrate that this translates to improved performance on several MIL datasets.

3.1. Supervised contrastive learning with pseudo labels

To address the class collision problem described in Section 2, we leverage a *supervised* contrastive learning approach (Pantazis et al., 2021; Dwibedi et al., 2021; Khosla et al., 2020) combined with the pseudo labels estimated by the aggregator. The goal is to learn a discriminative representation by pulling together the feature vectors corresponding to instances in the same class, and pushing apart those of instances with different classes. For a certain instance x, assume we have available two sets of instances $\mathcal{S}_x$ and $\mathcal{D}_x$, which we believe belong to the same class ($\mathcal{S}_x$) and to a different class ($\mathcal{D}_x$). The corresponding supervised contrastive loss corresponding to a single instance x is listed in Equation 2:

$$\mathcal{L}_{\text{sup}}(x) = \frac{1}{|\mathcal{S}_x|} \sum_{x_s \in \mathcal{S}_x} -\log \frac{\exp\left(f_\psi(x) \cdot f_\psi(x_s)/\tau\right)}{\sum_{x_s \in \mathcal{S}_x} \exp\left(f_\psi(x_i) \cdot f_\psi(x_s)/\tau\right) + \sum_{x_d \in \mathcal{D}_x} \exp\left(f_\psi(x) \cdot f_\psi(x_d)/\tau\right)}. \tag{2}$$

In Section 3.2, we explain how to select x, $\mathcal{S}_x$ and $\mathcal{D}_x$ to ensure that the corresponding pseudo labels are reliable. In Figure 2, we illustrate the intended effect of minimizing Equation 2. The careful reader may be wondering why one cannot just use the more standard supervised cross-entropy loss to train the feature extractor from the pseudo labels. In Section 4.3 we show that this leads to a substantially worse performance in the downstream MIL classification task.

3.2. Sampling via self-paced learning

A key consideration in the It2SCLR framework is how to select the subset of *query instances* on which to apply the supervised contrastive loss in equation 2. Our strategy is to exploit the bag labels and the instance class probabilities obtained from the aggregator. Let $\mathcal{X}^-_{\text{neg}}$ denote all instances within the negative bags. By definition of the MIL problem, we can safely assume that all instances in $\mathcal{X}^-_{\text{neg}}$ are negative. In contrast, positive bags contain both positive and negative instances. Let $\mathcal{X}^+_{\text{pos}}$ and $\mathcal{X}^-_{\text{pos}}$ denote the set of instances in positive bags with positive and negative pseudo labels respectively. During an initial warm-up lasting $T_{\text{warm-up}}$ epochs, we sample the query instances exclusively from $\mathcal{X}^-_{\text{neg}}$ to ensure that they are indeed all negative. For each such query instance, $\mathcal{S}_x$ is built by sampling and randomly augmenting instances from $\mathcal{X}^-_{\text{neg}}$, and $\mathcal{D}_x$ is built by sampling from $\mathcal{X}^+_{\text{pos}}$.

After the warm-up phase, we gradually incorporate query instances from the positive bags. In order to ensure reliability of the pseudo labels, we rank the instances in $\mathcal{X}^+_{\text{pos}}$ and $\mathcal{X}^-_{\text{pos}}$ according to the class probabilities obtained from the aggregator (which are also used to assign the pseudo labels in the first place). We denote the top $r\%$ instances with highest probabilities by $\mathcal{X}^+_{\text{pos}}(r)$ and $\mathcal{X}^-_{\text{pos}}(r)$ respectively. We sample the positive queries from $\mathcal{X}^+_{\text{pos}}(r)$, and the negative queries from $\mathcal{X}^-_{\text{neg}} \cup \mathcal{X}^-_{\text{pos}}(r)$ (the ratio between positive and negative queries is a hyperparameter). For a positive query x, we then sample the set of similar instances $\mathcal{S}_x$ from $\mathcal{X}^+_{\text{pos}}(r)$ and the set of different instances $\mathcal{D}_x$ from $\mathcal{X}^-_{\text{neg}} \cup \mathcal{X}^-_{\text{pos}}(r)$. For a negative query, $\mathcal{S}_x$ is sampled from $\mathcal{X}^-_{\text{neg}} \cup \mathcal{X}^-_{\text{pos}}(r)$ and $\mathcal{D}_x$ from $\mathcal{X}^+_{\text{pos}}(r)$. In order to exploit the improvement of the instance vectors during training, we gradually increase r to include more query instances from positive bags, which can be interpreted as a self-paced *easy-to-hard* learning scheme (Kumar et al., 2010; Jiang et al., 2014; Zou et al., 2018). Let t and T denote the current epoch, and the total number of epochs respectively. For $T_{\text{warmup}} < t \leq T$ we set:

$$r := r_0 + \alpha_r \left(t - T_{\text{warmup}}\right), \text{ where } \alpha_r = \frac{r_T - r_0}{T - T_{\text{warmup}}}, \tag{3}$$

where r_0 and r_T are hyperparameters. Details on tuning these hyperparameters are shown in Appendix A.3. As demonstrated in the graph on the right of Figure 1 (more metrics in Appendix B.1), the scheme indeed results in an improvement of the pseudo labels (and hence of the underlying feature vectors).

4. Experiments

In this section, we evaluate ItS2CLR on three real-world MIL datasets, described in Section 4.1. In Section 4.2 we show that ItS2CLR consistently outperforms approaches that use CSSL feature extraction by a substantial margin on all three datasets for different choices of aggregators. In Section 4.3 we show that ItS2CLR also outperforms alternative finetuning approches based on cross-entropy loss minimization and end-to-end training across a wide range of settings where the prevalence of positive instances and bag size vary.

4.1. Datasets

We evaluate the proposed framework on real-world datasets associated to different tasks relevant to cancer diagnostics. When training our models, we select the model with highest bag-level performance on the validation set and report the performance on a held-out test set. More information about the datasets, experimental setup, and implementation details can be found in Appendix A.

Camelyon16 (Bejnordi et al., 2017) is currently one of the main benchmark datasets for MIL (Li et al., 2021; Shao et al., 2021; Zhang et al., 2022). Its associated task is the detection of breast-cancer metastasis in 400 whole-slide histopathology images of lymph node sections. Each whole slide image (WSI) is paired with a binary label indicating the presence of cancer. Each WSI is divided into an average of 625 tiles, which correspond to individual instances. This dataset is particularly useful for benchmarking MIL models, because it has pixel-wise annotations indicating locations with cancer, from which we derive ground-truth instance-level labels.

TCGA-LUAD is a dataset extracted from The Cancer Genome Atlas (TCGA) (tcg), a landmark cancer genomics program. The associated task is to detect genetic mutations within 800 tumorous frozen whole-slide histopathology images from lung adenocarcinoma (LUAD). Detecting these mutations is important to determine treatment options in LUAD (Coudray et al., 2018; Fu et al., 2020). Similar to Camelyon16, each WSI is divided into an average of 633 tiles, which correspond to individual instances.

In the **Breast Ultrasound Dataset**, the task is to detect breast cancer in 28,914 B-mode breast ultrasound exams (Shen et al., 2021a). Each exam contains 4-70 images (18.8 images per exam on average), which correspond to individual instances, but only one exam-level label is available, indicating the presence of cancer. Additionally, a subset of images is annotated, which makes it possible to also evaluate instance-level performance.

4.2. Comparison with contrastive self-supervised learning

In this section, we compare the performance of ItS2CLR to a baseline that performs feature extraction based on the CSSL method SimCLR (Chen et al., 2020; Ciga et al., 2022).

Table 2: Bag-level AUC of ItS2CLR and a two-stage baseline using a SimCLR feature extractor and a MIL aggregator. Both models use the same aggregator (see Appendix A.2 for more details). ItS2CLR outperforms the baseline on all three datasets.

AUC ($\times 10^{-2}$)	Camelyon16	Breast Ultrasound	TCGA-LUAD mutation			
			EGFR	KRAS	STK11	TP53
SimCLR + Aggregator	85.38	80.79	67.51	68.79	70.40	62.15
ItS2CLR	**94.25**	**93.93**	**72.30**	**71.06**	**75.08**	**65.61**

Table 3: Bag-level AUC on Camelyon16 for ItS2CLR and different baselines for five aggregators. We retrain each aggregator 5 times to report the mean and standard deviation. All feature extractors are initialized using SimCLR. Ground-truth and cross-entropy (CE) finetuning further optimize the feature extractor using ground-truth instance-level labels and pseudo labels respectively. We also include ablated versions of ItS2CLR without iterative updates (w/o iter.), self-paced learning (w/o SPL) and both (w/o both), and an ablated version of CE finetuning without iterative updates (w/o iter).

AUC ($\times 10^{-2}$)	SimCLR (CSSL)	Ground-truth finetuning*	CE finetuning		ItS2CLR			
			w/o iter.	*iter.*	*w/o both*	*w/o iter.*	*w/o SPL*	*Full*
Max pooling	$86.69_{1.09}$	$98.25_{0.01}$	$85.48_{0.24}$	$88.05_{0.77}$	$85.38_{0.31}$	$91.96_{0.31}$	$90.85_{0.76}$	$\mathbf{94.69}_{0.07}$
Top-k pooling (Shen et al., 2021b)	$85.39_{1.20}$	$98.39_{0.05}$	$85.96_{0.45}$	$87.26_{0.42}$	$85.46_{0.21}$	$91.73_{0.42}$	$91.69_{0.28}$	$\mathbf{95.07}_{0.09}$
Attention-MIL (Ilse et al., 2018)	$79.49_{3.20}$	$99.06_{0.02}$	$88.50_{0.54}$	$90.46_{0.64}$	$85.21_{0.74}$	$93.13_{0.22}$	$86.20_{3.25}$	$\mathbf{94.45}_{0.05}$
DS-MIL (Li et al., 2021)	$85.38_{1.32}$	$98.65_{0.08}$	$87.01_{0.82}$	$90.38_{0.67}$	$85.08_{0.38}$	$91.69_{0.54}$	$88.29_{0.99}$	$\mathbf{94.25}_{0.07}$
Transformer (Chen & Krishnan, 2022)	$87.25_{0.59}$	$98.85_{0.25}$	$89.02_{0.54}$	$92.13_{1.07}$	$87.13_{0.71}$	$93.52_{0.49}$	$92.12_{0.68}$	$\mathbf{95.74}_{0.27}$

This approach has achieved state-of-the-art performance on multiple WSI datasets (Li et al., 2021). To ensure a fair comparison, we initialize the feature extractor in ItS2CLR also using SimCLR. Table 2 shows that ItS2CLR clearly outperforms the SimCLR baseline on all three datasets. The performance improvement is particularly significant in Camelyon16 where it achieves a bag-level AUC of 0.943, outperforming the baseline by a margin of 8.87%. ItS2CLR also outperforms an improved baseline reported by Li et al. (2021) with an AUC of 0.917, which uses higher resolution tiles than in our experiments (both 20x and 5x, as opposed to 5x only).

In order to perform a more exhaustive comparison of the features learned by SimCLR and ItS2CLR, we use them to train a range of popular MIL aggregators[1] to perform bag-level predictions on Camelyon16, including max pooling, top-k pooling (Shen et al., 2021b), attention-MIL pooling (Ilse et al., 2018), DS-MIL pooling (Li et al., 2021), and transformer (Chen & Krishnan, 2022) (see Appendix C for a detailed description). Table 3 shows that the ItS2CLR features outperform the SimCLR features by a larger margin for all aggregators, and is substantially more stable (the standard deviation of the AUC over multiple trials is lower).

An important secondary task in MIL for cancer prediction is instance-level localization. In Table 4, we report the instance-level AUC, F1 score, and Dice score of both ItS2CLR and the SimCLR-based baseline on Camelyon16. ItS2CLR again exhibits better performance. This is further illustrated in Figure 3, where we show a tumor localization map.

4.3. Comparison with alternative approaches

In Table 3,4,5, we compare ItS2CLR with additional alternative approaches. In what follows, we introduce these alternative approaches. In addition, we also analyze ItS2CLR's performance (Table 6) under different *witness rates* (fraction of positive instances in each positive bag). We create multiple witness rate settings by synthetically downsampling negative and positive instances in Camelyon16.

Finetuning with ground-truth instance labels provides an upper bound on the performance that can be achieved through feature improvement. ItS2CLR does not reach this gold standard, but substantially closes the gap.

Cross-entropy finetuning with pseudo labels, which we refer to as *CE finetuning*, consistently underperforms ItS2CLR when combined with different aggregators, except at high witness rates. We conjecture that this is due to the sensitivity of the cross-entropy loss to pseudo label noise.

Ablated versions of ItS2CLR where we do not apply iterative updates of the pseudo labels (w/o iter), or our self-paced learning scheme (w/o SPL) or both (w/o both) achieve substantially worse performance than the full approach. This

[1]To be clear, in this comparison the ItS2CLR features are learned using the DS-MIL aggregator, as described in Section 3, and then frozen.

Table 4: Comparison of instance-level performance for the models included in Table 3. All models use the DS-MIL aggregator. ItS2CLR achieves the best localization performance. Dice score is computed from a post-processed probability estimate described in Appendix B.2, which includes further details and results for other aggregators.

$(\times 10^{-2})$	SimCLR (CSSL)	Ground-truth finetuning	CE finetuning		ItS2CLR			
			w/o iter.	*iter.*	*w/o both*	*w/o iter.*	*w/o SPL*	*Full*
AUC	94.01	97.94	95.69	96.06	95.13	95.90	96.12	**96.72**
F1-score	84.49	88.01	86.94	86.93	86.74	87.45	**87.95**	87.47
AUPRC	86.57	86.13	89.26	89.39	88.30	89.51	90.00	**91.12**
Dice (*)	31.79	62.17	49.11	49.41	43.74	51.70	53.03	**57.86**

Table 5: Comparison with models trained end-to-end models, which are initialized with the same pretrained weights as ItS2CLR, and use the same aggregator.

	Camelyon16 (downsampled synthetic)			Breast Ultrasound			
	Bag AUC	Instance AUC	Instance F-score	Bag AUC	Bag AUPRC	Instance AUC	Instance AUPRC
End-to-end	66.71	78.32	55.71	91.26	58.73	82.11	31.31
ItS2CLR	**88.65**	**95.58**	**87.01**	**93.93**	**70.30**	**88.63**	**43.71**

indicates that both of these ingredients are critical in learning discriminative features.

End-to-end training is often computationally infeasible in practice for medical applications. We compare ItS2CLR to end-to-end models on a downsampled version of Camelyon16 (see Appendix A.4) and on the breast ultrasound dataset. For fair comparison, all end-to-end models use the same CSSL-pretrained weights and aggregator as ItS2CLR. Table 5 shows that ItS2CLR achieves better instance- and bag-level performance than end-to-end training. The analysis in Appendix B.3 shows that end-to-end can overfit quite fast under large-bag-size and weak supervision settings.

5. Related work

Contrastive Self-Supervised Learning Contrastive learning methods have become popular in unsupervised representation learning, achieving the state-of-the-art self-supervised learning performance for natural images (Chen et al., 2020; He et al., 2020; Grill et al., 2020a; Caron et al., 2020; Zbontar et al., 2021; Caron et al., 2021). These methods have also shown promising results in medical imaging (Li et al., 2021; Ciga et al., 2022; Azizi et al., 2021; Kaku et al., 2021; Zhu et al., 2022). Recently, Li et al. (2021) applied SimCLR (Chen et al., 2020) to extract instance-level features for WSI MIL tasks and achieved the state-of-the-art performance. However, Arora et al. (2019) point out the potential issue of class collision, i.e. that some negative pairs may actually have the same class. Prior works on alleviating class collision problem includes reweighting the negative and positive terms with class ratio (Chuang et al., 2020), pulling closer additional similar pairs (Dwibedi et al., 2021), and avoiding pushing apart negatives that are inferred to belong to the same class based on a similarity metric (Zheng et al., 2021). In contrast, we propose a framework that leverages information from the bag-level labels to iteratively resolve the class collision problem.

There also exist non-contrastive alternatives that avoid introducing negative pairs (*e.g.* BYOL (Grill et al., 2020b) and SimSiam (Chen & He, 2021)). However, Wang et al. (2021) report that removing the negatives can make different object categories overlap and results in under-clustering, which limits the model's ability to learn discriminative features.

MIL Aggregators Traditionally, non-learnable pooling such as mean-pooling and max-pooling were commonly used in MIL (Feng & Zhou, 2017; Pinheiro & Collobert, 2015). More recent methods parameterize the aggregator using neural networks that employ attention mechanisms (Ilse et al., 2018; Li et al., 2021). Most recently, transformer-based aggregators (Chen & Krishnan, 2022; Shao et al., 2021) have achieved impressive results. This research direction is complementary with our proposed approach, which focuses on obtaining better instance representations and can be combined with different types of aggregators.

6. Discussion

Our results show several limitations of our approach. First, it does not outperform a cross-entropy-based baseline at very high witness rates, suggesting that it is mostly suitable for low witness rates scenarios (however, it is worth

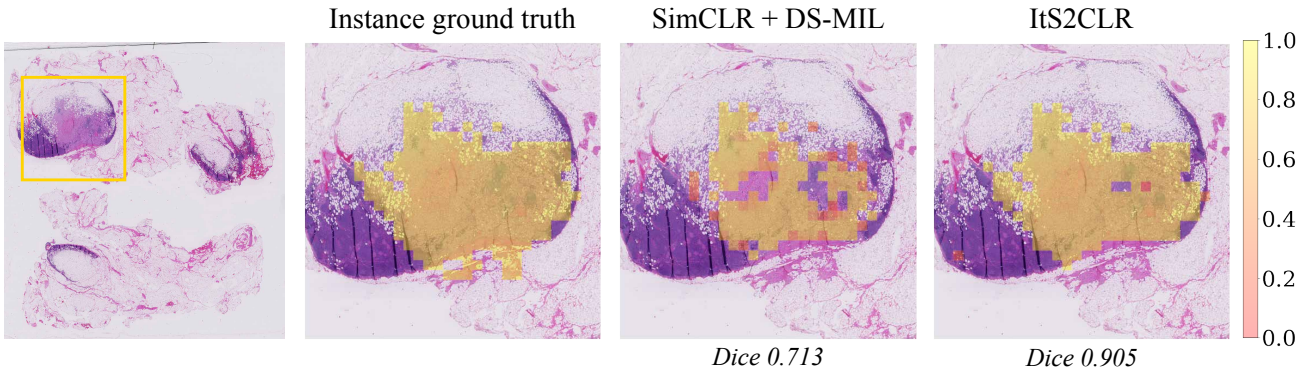

Figure 3: Tumor localization for a histopathology slide from the Camelyon16 test set. Instance predictions are generated by the instance classifier of the DS-MIL trained on extracted instance-level features. Appendix B.4 shows additional results.

Table 6: Bag-level AUC on Camelyon16 across different witness rates (WR). All methods use the DS-MIL aggregator for a fair comparison. When the WR is low, ItS2CLR outperforms CE finetuning by a large margin. As the WR increases, CE finetuning becomes more effective.

Downsampled Instances	WR (%)	SimCLR (CSSL)	CE finetuning iterative	ItS2CLR	Finetuning w. instance GT
5% Neg.	71.2	94.52	**98.55**	97.58	99.11
10% Neg.	45.0	93.70	**97.88**	96.15	99.18
40% Neg.	23.5	90.38	93.32	**95.40**	97.68
Original	10.9	85.38	90.38	**94.25**	98.65
50% Pos.	5.8	82.47	86.96	**88.52**	91.81
33% Pos.	4.1	78.21	80.56	**86.02**	88.01

noting that this is the regime that is more commonly encountered in medical applications such as cancer diagnostics). In addition, there is a performance gap between our method and fine-tuning using instance-level ground truths, which suggests there is further room for improvement. Finally, we only evaluate our models on medical imaging datasets. Future research could focus on extending our framework to other applications, such as weakly-labelled video classification (Luo et al., 2020; Feng et al., 2021).

We evaluate our method on cancer detection using histopathology and ultrasound images. Our experiments suggest that the proposed method has the potential to help pathologists and other clinicians in the analysis of medical images, which could result in improved diagnosis and reduced costs. While our evaluation shows that our method can improve localization performance without access to instance-level labels, only using bag labels may increase the chance of over-fitting to potential outliers that may lead to erroneous results. Further study on the reliability of cancer localization with bag-level supervision is therefore recommended.

7. Conclusion

In this work, we investigate how to improve the feature extractor in two-stage MIL models. We identify a limitation of the existing contrastive self-supervised learning, namely, that it fails to learn distriminative features in class-imbalanced MIL problems. We provide analysis on InfoNCE loss showing that class collision is the fundamental cause for this problem. To solve it, we propose a novel framework that iteratively refines the features with pseudo labels estimated by the aggregator. Our method outperforms contrastive SSL pretraining on three medical datasets. The consistent performance over a broad range of downstream aggregators, witness rates, and bag sizes demonstrates its generalization capability.

References

The cancer genome atlas program. https://www.cancer.gov/tcga.

Arora, S., Khandeparkar, H., Khodak, M., Plevrakis, O., and Saunshi, N. A theoretical analysis of contrastive unsupervised representation learning. *arXiv preprint*

arXiv:1902.09229, 2019.

Ash, J. T., Goel, S., Krishnamurthy, A., and Misra, D. Investigating the role of negatives in contrastive representation learning. *arXiv preprint arXiv:2106.09943*, 2021.

Azizi, S., Mustafa, B., Ryan, F., Beaver, Z., Freyberg, J., Deaton, J., Loh, A., Karthikesalingam, A., Kornblith, S., Chen, T., et al. Big self-supervised models advance medical image classification. In *CVPR*, 2021.

Bejnordi, B. E., Veta, M., Van Diest, P. J., Van Ginneken, B., Karssemeijer, N., Litjens, G., Van Der Laak, J. A., Hermsen, M., Manson, Q. F., Balkenhol, M., et al. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *Jama*, 318(22):2199–2210, 2017.

Campanella, G., Hanna, M. G., Geneslaw, L., Miraflor, A., Werneck Krauss Silva, V., Busam, K. J., Brogi, E., Reuter, V. E., Klimstra, D. S., and Fuchs, T. J. Clinical-grade computational pathology using weakly supervised deep learning on whole slide images. *Nature medicine*, 25(8): 1301–1309, 2019.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021.

Chen, R. J. and Krishnan, R. G. Self-supervised vision transformers learn visual concepts in histopathology, 2022. URL https://arxiv.org/abs/2203.00585.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations, 2020.

Chen, X. and He, K. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15750–15758, 2021.

Chikontwe, P., Kim, M., Nam, S. J., Go, H., and Park, S. H. Multiple instance learning with center embeddings for histopathology classification. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 519–528. Springer, 2020.

Chuang, C.-Y., Robinson, J., Lin, Y.-C., Torralba, A., and Jegelka, S. Debiased contrastive learning. *Advances in neural information processing systems*, 33:8765–8775, 2020.

Ciga, O., Xu, T., and Martel, A. L. Self supervised contrastive learning for digital histopathology. *Machine Learning with Applications*, 7:100198, 2022.

Clark, A. Pillow (pil fork) documentation, 2015. URL https://buildmedia.readthedocs.org/media/pdf/pillow/latest/pillow.pdf.

Coudray, N., Ocampo, P. S., Sakellaropoulos, T., Narula, N., Snuderl, M., Fenyö, D., Moreira, A. L., Razavian, N., and Tsirigos, A. Classification and mutation prediction from non–small cell lung cancer histopathology images using deep learning. *Nature Medicine*, 24(10):1559–1567, 2018. doi: 10.1038/s41591-018-0177-5. URL https://doi.org/10.1038/s41591-018-0177-5.

Courtiol, P., Tramel, E. W., Sanselme, M., and Wainrib, G. Classification and disease localization in histopathology using only global labels: A weakly-supervised approach. *arXiv preprint arXiv:1802.02212*, 2018.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P., and Zisserman, A. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9588–9597, 2021.

Feng, J. and Zhou, Z.-H. Deep miml network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

Feng, J., Hong, F., and Zheng, W. MIST: multiple instance self-training framework for video anomaly detection. *CoRR*, abs/2104.01633, 2021. URL https://arxiv.org/abs/2104.01633.

Fu, Y., Jung, A. W., Torne, R. V., Gonzalez, S., Vöhringer, H., Shmatko, A., Yates, L. R., Jimenez-Linan, M., Moore, L., and Gerstung, M. Pan-cancer computational histopathology reveals mutations, tumor composition and prognosis. *Nature Cancer*, 1(8):800–810, 2020. doi: 10.1038/s43018-020-0085-8. URL https://doi.org/10.1038/s43018-020-0085-8.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., Piot, B., kavukcuoglu, k., Munos, R., and Valko, M. Bootstrap your own latent - a new approach to self-supervised learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 21271–21284. Curran Associates,

Inc., 2020a. URL https://proceedings.neurips.cc/paper/2020/file/f3ada80d5c4ee70142b17b8192b2958e-Paper.pdf.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020b.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

Ilse, M., Tomczak, J., and Welling, M. Attention-based deep multiple instance learning. In *International conference on machine learning*, pp. 2127–2136. PMLR, 2018.

Jiang, L., Meng, D., Yu, S.-I., Lan, Z., Shan, S., and Hauptmann, A. Self-paced learning with diversity. *Advances in neural information processing systems*, 27, 2014.

Kaku, A., Upadhya, S., and Razavian, N. Intermediate layers matter in momentum contrastive self supervised learning. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=M5j42PvY65V.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., and Krishnan, D. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020.

Kumar, M., Packer, B., and Koller, D. Self-paced learning for latent variable models. *Advances in neural information processing systems*, 23, 2010.

Lerousseau, M., Vakalopoulou, M., Classe, M., Adam, J., Battistella, E., Carré, A., Estienne, T., Henry, T., Deutsch, E., and Paragios, N. Weakly supervised multiple instance learning histopathological tumor segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 470–479. Springer, 2020.

Li, B., Li, Y., and Eliceiri, K. W. Dual-stream multiple instance learning network for whole slide image classification with self-supervised contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14318–14328, 2021.

Lu, M. Y., Chen, R. J., and Mahmood, F. Semi-supervised breast cancer histology classification using deep multiple instance learning and contrast predictive coding (conference presentation). In *Medical Imaging 2020: Digital Pathology*, volume 11320, pp. 113200J. International Society for Optics and Photonics, 2020.

Lu, M. Y., Williamson, D. F., Chen, T. Y., Chen, R. J., Barbieri, M., and Mahmood, F. Data-efficient and weakly supervised computational pathology on whole-slide images. *Nature biomedical engineering*, 5(6):555–570, 2021.

Luo, Z., Guillory, D., Shi, B., Ke, W., Wan, F., Darrell, T., and Xu, H. Weakly-supervised action localization with expectation-maximization multi-instance learning. *CoRR*, abs/2004.00163, 2020. URL https://arxiv.org/abs/2004.00163.

Pantazis, O., Brostow, G. J., Jones, K. E., and Mac Aodha, O. Focus on the positives: Self-supervised learning for biodiversity monitoring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10583–10592, 2021.

Pinheiro, P. O. and Collobert, R. From image-level to pixel-level labeling with convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1713–1721, 2015.

Shao, Z., Bian, H., Chen, Y., Wang, Y., Zhang, J., Ji, X., et al. Transmil: Transformer based correlated multiple instance learning for whole slide image classification. *Advances in Neural Information Processing Systems*, 34, 2021.

Shen, Y., Shamout, F. E., Oliver, J. R., Witowski, J., Kannan, K., Park, J., Wu, N., Huddleston, C., Wolfson, S., Millet, A., et al. Artificial intelligence system reduces false-positive findings in the interpretation of breast ultrasound exams. *Nature communications*, 12(1):1–13, 2021a.

Shen, Y., Wu, N., Phang, J., Park, J., Liu, K., Tyagi, S., Heacock, L., Kim, S. G., Moy, L., Cho, K., et al. An interpretable classifier for high-resolution breast cancer screening images utilizing weakly supervised localization. *Medical image analysis*, 68:101908, 2021b.

Vahadane, A., Peng, T., Sethi, A., Albarqouni, S., Wang, L., Baust, M., Steiger, K., Schlitter, A. M., Esposito, I., and Navab, N. Structure-preserving color normalization and sparse stain separation for histological images. *IEEE Transactions on Medical Imaging*, 35(8): 1962–1971, 2016. doi: 10.1109/TMI.2016.2529665.

Wang, G., Wang, K., Wang, G., Torr, P. H., and Lin, L. Solving inefficiency of self-supervised representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9505–9515, 2021.

Xu, G., Song, Z., Sun, Z., Ku, C., Yang, Z., Liu, C., Wang, S., Ma, J., and Xu, W. Camel: A weakly supervised learning framework for histopathology image segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10682–10691, 2019.

Zbontar, J., Jing, L., Misra, I., LeCun, Y., and Deny, S. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230*, 2021.

Zhang, H., Meng, Y., Zhao, Y., Qiao, Y., Yang, X., Coupland, S. E., and Zheng, Y. Dtfd-mil: Double-tier feature distillation multiple instance learning for histopathology whole slide image classification. *arXiv preprint arXiv:2203.12081*, 2022.

Zheng, M., Wang, F., You, S., Qian, C., Zhang, C., Wang, X., and Xu, C. Weakly supervised contrastive learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10042–10051, 2021.

Zhu, W., Fernandez-Granda, C., and Razavian, N. Interpretable prediction of lung squamous cell carcinoma recurrence with self-supervised learning, 2022. URL https://arxiv.org/abs/2203.12204.

Zou, Y., Yu, Z., Kumar, B., and Wang, J. Domain adaptation for semantic segmentation via class-balanced self-training. *arXiv preprint arXiv:1810.07911*, 2018.

Appendix for "Multiple Instance Learning via Iterative Self-Paced Supervised Contrastive Learning"

The appendix is organized as follows:

- In Appendix A, we include additional descriptions of the datasets (Appendix A.1), implementation details (Appendix A.2) and instructions on how to transform the Camelyon16 dataset for the additional experiments (Appendix A.4). In Appendix A.3, we provide a description of the hyperparameter selection process and report an ablation study on the Camelyon16 dataset to evaluate the sensitivity of our approach to the choice of hyperparameters.
- In Appendix B, we include additional results. In Appendix B.1, we report additional metrics to compare the pseudo label quality. In Appendix B.2, we show additional results for instance-level performance. In Appendix B.3, we report additional comparisons with end-to-end methods. In Appendix B.4, we provide additional examples of tumor localization maps.
- In Appendix C, we provide the formulation and implementation details for the different MIL aggregators used in our study.

License of the assets

Licence for the code

We used the publicly available code for DS-MIL (Li et al., 2021) https://github.com/binli123/dsmil-wsi and Attention-based MIL pooling https://github.com/AMLab-Amsterdam/AttentionDeepMIL. Both are published under the MIT License https://opensource.org/licenses/MIT

Licence for the datasets

For the Camelyon16 dataset, we follow the data use agreement at https://camelyon16.grand-challenge.org.

For the TCGA dataset, we follow the data use agreement at https://www.cancer.gov/about-nci/organization/ccg/research/structural-genomics/tcga

For the Breast Ultrasound dataset, we follow the instructions in the paper that introduced this dataset (Shen et al., 2021a).

A. Experiments

A.1. Dataset

Camelyon16 Camelyon16 is a public dataset for detection of metastasis in breast cancer. This dataset consists of 271 training and 129 test whole slide images (WSI), which are further divided into roughly 3.2 million patches at 20× magnification and 0.25 million patches at 5× magnification. On average, at 20× and 5× magnification each slide contains approximately 8,000 and 625 patches respectively. Each WSI is paired with pixel-level annotations indicating the position of tumors (if any are present). We ignore the pixel-level annotations during training and consider only slide-level labels (i.e. the slide is considered positive if it contains any annotated tumor regions). As a result, positive bags contain mixtures of patches with tumors and patches with healthy tissue. Negative bags contain only patches with healthy tissue. The ratio between positive and negative patches in this dataset is highly imbalanced. Only a small fraction of patches (less than 10%) in the positive slides contains tumor.

TCGA-LUAD TCGA for Lung Adenocarcinoma (LUAD) is a subset of TCGA (The Cancer Genome Atlas), a landmark cancer genomics program. It consists of 800 tumorous frozen whole-slide histopathology images and the corresponding genetic mutation status. Each WSI is paired with binary labels indicating whether each gene is mutated or wild type. In this experiment, we build MIL models to detect four mutations - EGFR, KRAS, STK11, and TP53, which are sensitizing mutations that can impact treatment options in LUAD (Coudray et al., 2018; Fu et al., 2020). We split the data randomly into training, validation and test sets so that each patient will appear in only one of the subsets. After splitting the data, 477 images are in the training set, 96 images are in the validation set, and 227 images are in the test set.

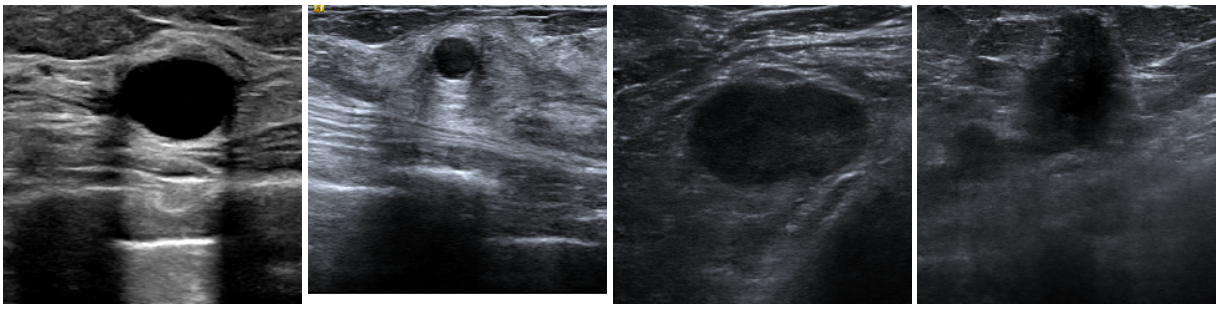

Figure 4: Example breast ultrasound images. The first two images contain a benign lesion. The second and third contain a malignant lesion. In all ultrasound images, the center object of circular shape corresponds to the lesion of interest.

Breast Ultrasound dataset The Breast Ultrasound Dataset includes 28,914 ultrasound exams (Shen et al., 2021a). An exam is labeled as cancer-positive if there is a pathology-confirmed malignant finding associated with this exam. In this dataset, 5593 exams are cancer-positive. On average, each exam contains approximately 13 images. Patients in the dataset were randomly divided into a training set (60%), a validation set (10%), and test set (30%). Each patient was included in only one of the three sets. We show 5 example breast ultrasound images in Figure 4.

A.2. Implementation Details

All experiments were conducted on NVIDIA RTX8000 GPUs and NVIDIA V100 GPUs. For all models, we perform model selection during training based on bag-level AUC evaluated on the validation set.

Camelyon16 We follow the same preprocessing and pretraining steps as Li et. al. (Li et al., 2021). To preprocess the slides, we cropped the slides into tiles at 5x magnification, filtered out tiles that do not contain enough tissues (average saturation < 30), and resized the images into a resolution of 224 x 224 pixels. Resizing was performed using the Pillow package (Clark, 2015) with default settings (nearest neighbor sampling).

We pretrain the feature extractor, ResNet18 (He et al., 2016), with SimCLR (Chen et al., 2020) for a maximum of 600 epochs. Each patch is represented by a 512-dimensional vector. We set the batch size at 512 and temperature at 0.5. We use SGD with the learning rate of 0.03, weight decay of 0.0001, and cosine annealing scheduler. We also train a MIL aggregator using the instance features extracted by the feature extractor in order to monitor the bag AUC of the downstream task on the validation set. During finetuning with Its2CLR, we fine-tune the feature extractor for a maximum of 50 epochs. The batch size is set to 512, and the learning rate is set to 10^{-2}. At the feature extractor training stage, we apply random data augmentation to each instance, including:

- Random ($p = 0.8$) color jittering: brightness, contrast, and saturation factors are uniformly sampled from $[0.2, 1.8]$, hue factor is uniformly sampled from $[-0.2, 0.2]$;
- Random gray scale ($p = 0.2$);
- Random Gaussian blur with kernel size of 0.06 times the size of an image;
- Random horizontal/vertical flipping with 0.5 probability.

When training the DS-MIL aggregator, we follow the settings in (Li et al., 2021). We use the Adam optimizer during training. Since each bag may contain different a number of instances, we follow (Li et al., 2021) and set the batch size to just one bag. We train each model for a maximum of 350 epochs. We use an initial learning rate of 2×10^{-4}, and use the StepLR scheduler to reduce the learning rate by 0.5 every 75 epochs. Details on the hyperparameters used for training the aggregator are in Appendix C.

TCGA-LUAD To preprocess the slides, we cropped them into tiles at 10x magnification, filtered out the background tiles that do not contain enough tissues (when average saturation is less than 30), and resized the images into the resolution of 224 x 224 pixels. Resizing was performed using the Pillow package (Clark, 2015) with nearest neighbor sampling. These tiles were color-normalized to match the Vahadane method (Vahadane et al., 2016).

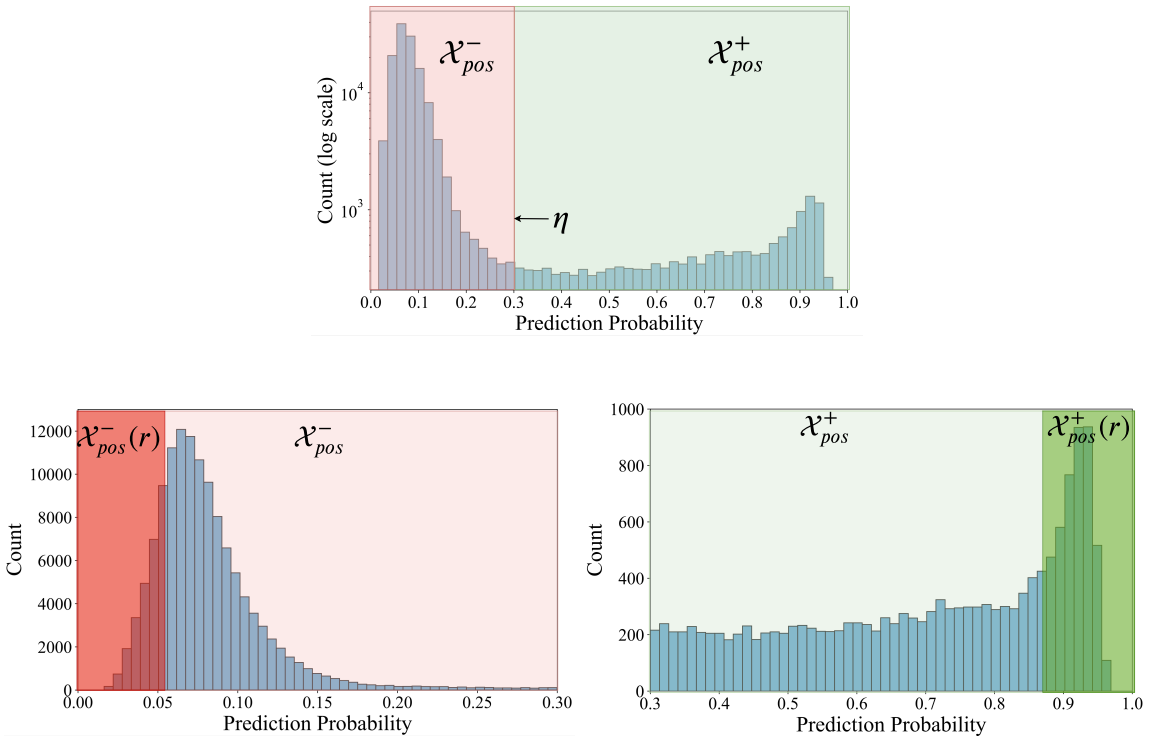

Figure 5: Illustration of our partitioning of the instances from positive bags in Section 3.2 based on the predicted probability of the instance classifier in ItS2CLR. **Top**: $\mathcal{X}^+_{\text{pos}}$ and $\mathcal{X}^-_{\text{pos}}$ are partitioned according to the thresholding parameter η. **Bottom**: The distribution of instance scores for instances with negative pseudo labels (left) and negative pseudo labels (right). A threshold r is symmetrically applied on both distributions so that the top $r\%$ instances with the lowest and highest scores are treated as truly negative or positive respectively. We use $\mathcal{X}^-_{\text{pos}}(r)$ and $\mathcal{X}^+_{\text{pos}}(r)$ to denote the set of instances that are deemed truly negative and positive respectively. During training, as the quality of the pseudo labels improves, we can increase r to incorporate more samples in these sets.

To train the feature extractor, we perform the same process as for Camelyon16.

We also use DS-MIL (Li et al., 2021) as the aggregator. When training the aggregator, we resample the ratio of positive and negative bags to keep the class ratio balanced. We train the aggregator for a maximum of 100 epochs using the Adam optimizer with the learning rate set to 2×10^{-4} and reduce the learning rate by 0.5 every 50 epochs.

Breast Ultrasound We follow the same preprocessing steps as Shen et. al. (Shen et al., 2021a). All images were resized to 224 x 224 pixels using bilinear interpolation. We used ResNet18 (He et al., 2016) as the feature extractor and pretrained it using SimCLR (Chen et al., 2020) for 100 epochs. We adopt the same pretraining approach as for Camelyon16. We used the Instance Attention-MIL as an aggregator (Ilse et al., 2018). Given a bag of images $x_1, ..., x_k$ and a feature extractor f, the aggregator first computes instance-level predictions $\hat{y}_i$ for each image x_i. It then calculates an attention score $\alpha_i \in [0, 1]$ for each image x_i using its feature vectors $f(x_i)$. Lastly, the bag-level prediction is computed as the average instance prediction weighted by the attention score $\hat{y} = \sum_i^k \alpha_i \hat{y}_i$. To optimize the aggregator, we trained it using Adam with a learning rate set to 10^{-3} for a maximum of 350 epochs.

A.3. Hyperparameters of Training the Feature Extractor in ItS2CLR

Hyperparameter tuning The hyperparameters of the proposed method include: the initial threshold used for binarization of the prediction to produce pseudo labels $\eta \in [0.1, 0.9]$, the proportion of positive queries sampled $p_+ \in [0.05, 0.5]$, the initial ratio $r_0 \in [0.01, 0.7]$ and the final ratio $r_T \in [0.2, 0.8]$ of in the self-paced sampling scheme. For Camelyon16, we obtain the highest bag-level validation AUC using the following hyperparameters: $\eta = 0.3$, $p_+ = 0.2$, $r_0 = 0.2$ and $r_T = 0.8$. We use the feature extractor trained under this setting in Tables 2, 3 and 4. The complete list of hyperparameters

in different experiments is reported in Table 7.

Table 7: ItS2CLR hyperparameters used in our experiments.

	Camelyon16	Breast Ultrasound	TCGA-LUAD mutation			
			EGFR	KRAS	STK11	TP53
η	0.3	0.3	0.3	0.5	0.3	0.5
r_0	0.2	0.2	0.2	0.2	0.2	0.2
r_T	0.8	0.8	0.8	0.8	0.8	0.8
p_+	0.2	0.2	0.5	0.2	0.2	0.2

Sensitivity analysis We conduct the sensitivity analysis for each hyperparameter on Camelyon 16, and observe a robust performance over a range of hyperparameter values.

- *Threshold* η: The choice of η influences the instance-level pseudo labels. As shown in Figure 5, the outputs of the instance-level classifier are mostly close to 0 or 1, so the pseudo labels do not dramatically vary for a wide range of η. We conducted a small ablation experiment on the importance of η. Figure 6 (left) shows that ItS2CLR is quite robust to the value of η, except for some extreme values. If η is too small (e.g. 0.1), it can introduce a significant number of false positives. If η is too large (e.g. 0.8), it can mistakenly exclude some useful positive samples, causing a drop in the performance. In the main paper, since negative instances are more prevalent than positive instances, a threshold of 0.3 (less than 0.5) can increase the recall for the positive instances.

- *Sampling ratio of query instance over pseudo labels*: We use p_+ to denote the percentage of positive query instances used during the contrastive learning stage. Figure 6 (right) shows that it is desirable to choose a relatively small p_+. Since there are far fewer positive instances than negative instance, keeping the ratio of positive queries low can avoid repetitively sampling from a limited number of positive instances. Also, since the negative instance set $\mathcal{X}_{\text{neg}}^-$ is clean, there is more label noise among the positive pseudo labels.

- *The initial rate* r_0 *and final rate* r_T *for the self-paced sampling scheduler*: Figure 7 shows that ItS2CLR is also generally robust to the r_0 and r_T. However, extremely large initial rate r_0 (high confidence in the pseudo labels) may introduce more noise during the training and hurt the performance. Conversely, extremely small r_T (low confidence in the pseudo labels) may prohibit the model from using more data, also hurting performance.

- *Sampling during warm-up:* During the warmup phase, we sample query instances from $\mathcal{X}_{\text{neg}}^-$. An alternative choice can be sampling we sample the query instance from $\mathcal{X}_{\text{pos}}^+$ and the corresponding set $\mathcal{D}_x$ from $\mathcal{X}_{\text{neg}}^-$. However, our experiments show that the resulting bag-level AUC drops to 90.91% under this setting, which is significantly lower than 94.25% by the proposed method. This comparison demonstrates the importance of using clean negative instances as query images during warmup.

A.4. Experiments on Synthetic Versions of Camelyon16

Simulation of witness rates (WR)

Since the ground truth instance-level labels are available for Camelyon16, we can conduct experiments on synthetic versions of the dataset to study the impact of the prevalence of positive instances on the performance of the proposed approach and the baselines. Section 2 describes how the performance of CSSL is affected by low *witness rates* (fraction of positive instances in each positive bag). To study the robustness of the proposed framework to the witness rate in the data, we increase or decrease the witness rate of Camelyon16 by randomly dropping negative or positive instances within each bag respectively. The percentage of retained instances and the resulting witness rates are reported in Table 6.

Downsampled version of Camelyon16 for end-to-end training

In order to enable end-to-end training we downsample each bag in Camelyon16 to around 500 instances so that it fits in the memory of a GPU. To achieve this, we divide the large bag into smaller bags while keeping the witness rate of each sub-bag at a similar level as the original bag. In more detail, if the bag size is smaller or equal to 500, we keep the original bag. If the

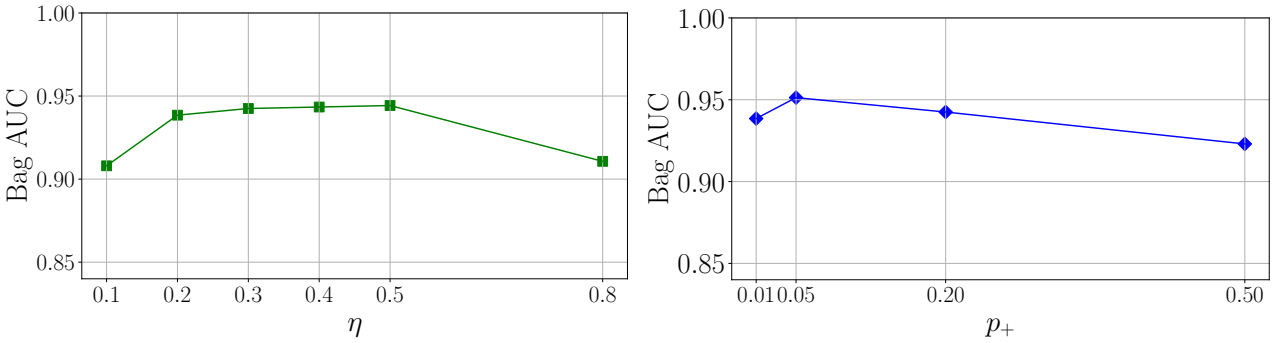

Figure 6: Sensitivity analysis for the threshold η and the ratio of positive pseudo labels used as query images p_+ on the Camelyon16 dataset.

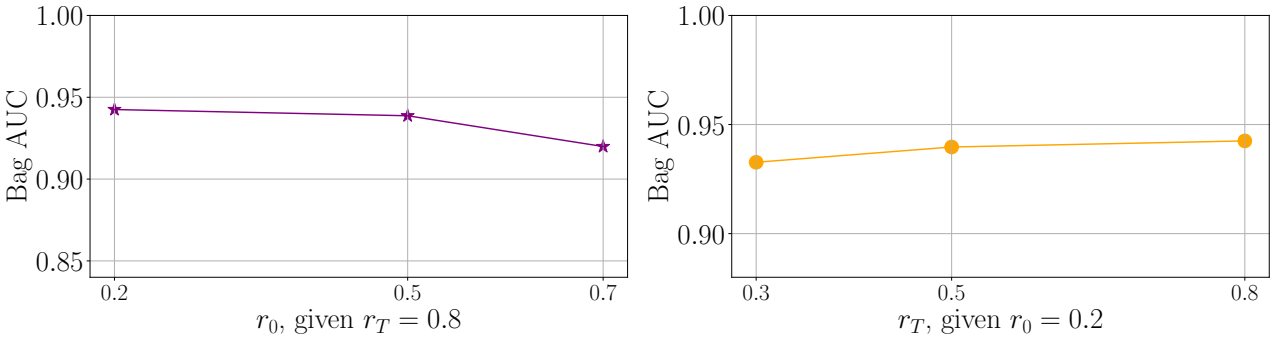

Figure 7: Sensitivity analysis for the hyperparameters r_0 and r_T of the proposed self-paced learning scheme on Camelyon16.

bag size is greater than 500, we divide the bag into several sub-bags such that each sub-bag contains approximately 500 instances. After that, we randomly divide the negative instances within the original bag into desired number of sub-bags evenly. For the positive bag, we randomly partition the positive instances within the original bag into desired number of sub-bags evenly as well. If the number of positive instances within that positive bag is smaller than the desired number of sub-bags, we correct the number of sub-bags to be the number of positive instances. We then combine the positive instances and the negative instances to form sub-bags. This ensures that the bag-level label is correct and the witness rate for each positive sub-bag remains similar to the original bag.

B. Additional Results

We present here additional results to supplement those presented in Section 4.

B.1. Learning Curves

F1-Score plot corresponding to Figure 1: In Figure 8, we show the max F1 score curve corresponding to the right side of Figure 1. This plot confirms the importance of self-paced learning and iterative updating in ItS2CLR.

Instance pseudo label AUC comparison with cross-entropy finetuning: Figure 9 compares ItS2CLR with an alternative approach that finetunes the feature extractor using cross-entropy (CE) loss on the Camelyon16 dataset. Without iterative updating, CE finetuning rapidly overfits to the noise in the pseudo labels. Iterative updating prevents this to some extent, but does not match the performance of ItS2CLR, which produces increasingly accurate pseudo labels as the iterations proceed.

B.2. Instance-level Evaluation

In order to evaluate instance-level performance, we report classification metrics including AUC, F1-score, AUPRC and Dice score for localization.

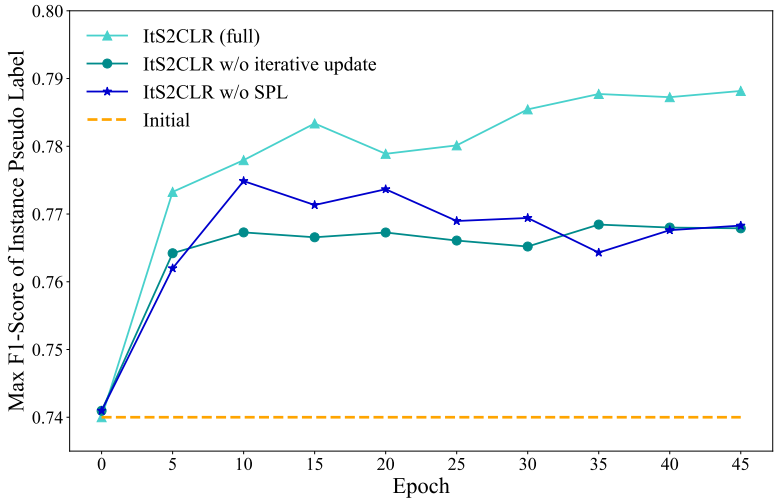

Figure 8: Max F1 score comparison. ItS2CLR updates the features iteratively based on a subset of the pseudo labels that is selected according to the self-paced learning (SPL) strategy. On Camelyon16, this gradually improves the accuracy of the pseudo labels in terms of instance-level max F1 score. Both the iterative updates and SPL are important to achieve this.

Table 8: Comparison of instance-level performance for the models in Table 3, using a max pooling aggregator.

$(\times 10^{-2})$	SimCLR (CSSL)	Ground-truth finetuning	CE finetuning		ItS2CLR			
			w/o iter.	*iter.*	*w/o both*	*w/o iter.*	*w/o SPL*	*Full*
AUC	91.53	97.58	93.17	94.48	92.69	94.55	94.43	**96.25**
F1-score	78.45	88.24	85.26	85.83	**86.77**	86.05	87.52	86.75
AUPRC	79.94	85.50	85.79	86.73	85.50	84.54	86.80	**89.99**
Dice	31.21	63.01	43.90	44.76	46.57	55.30	52.55	**57.82**

The Dice score is defined as follows:

$$\text{Dice} = \frac{2\sum_i y_i p_i}{\sum_i y_i + \sum_i p_i}, \tag{4}$$

where y_i and p_i are the ground truth and predicted probability for the ith sample. It penalizes the prediction with low confidence. The prediced probability is computed from the output of the MIL model s_i via linear scaling:

$$p_i = \sigma\left(as_i + b\right), \tag{5}$$

where $a \in [-5, 5]$ and $b \in [0.1, 10]$ are chosen to maximize the Dice score on the validation set.

Max pooling aggregator: In Table 4, we show that out model achieves better weakly supervised localization performance compared to other methods when DS-MIL is used as the aggregator. In Table 8, we show that the same conclusion holds for an aggregator based on max-pooling.

Linear evaluation: In Table 9, we report results obtained by training a logistic regression model using the features obtained from the same approaches in Table 4, following a standard linear evaluation pipeline in representation learning (Chen et al., 2020). ItS2CLR again achieves the best instance-level performance. We also produce bag-level predictions using the maximum output of the linear classifier for each bag, which again showcases that instance-level performance results in superior bag-level classification.

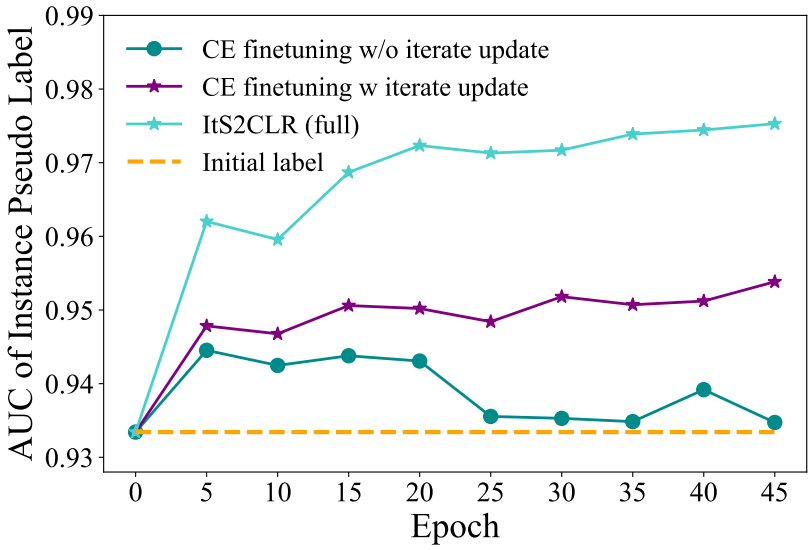

Figure 9: Comparison of instance-level pseudo label quality between ItS2CLR and an alternative approach that finetunes the feature extractor using cross-entropy (CE) loss on the Camelyon16 dataset. Iterative updating improves performance for CE finetuning, but ItS2CLR produces more accurate pseudo labels.

B.3. Comparison with End-to-end Training

In this section we provide additional results to complement Table 5, where ItS2CLR is compared to end-to-end models. The end-to-end training is conducted with the same aggregators for each dataset as described in Section 4 and Appendix A.2.

Camelyon16 Figure 10 shows that an end-to-end model trained on the downsampled version of Camelyon16 described in Section A.4 rapidly overfits when trained from scratch and from SimCLR-pretrained weights. The two-stage model, on the other hand, is less prone to overfitting. Table 10 shows that the two-stage learning pipeline outperforms end-to-end training, and is in turn outperfomed by ItS2CLR.

Breast Ultrasound dataset Table 11 shows that for the breast-ultrasound dataset end-to-end training outperforms the SimCLR+Aggregator baseline, but is outperformed by ItS2CLR.

B.4. Tumor Localization Maps

Figure 11 provides additional tumor localization maps.

C. MIL Aggregators

C.1. Formulation of MIL Aggregators

In this section, we describe the different MIL aggregators benchmarked in Section 4.2 and Table 3.

Let $\mathcal{B}$ denote a collection of sets of feature vectors in $\mathbb{R}^d$. The bags of extracted features in the dataset are denoted by $\{H_b\}_{b=1}^{B} \subset \mathcal{B}$. An aggregator is defined as a function $g : \mathcal{B} \to [0, 1]$ mapping bags of extracted features to a score in $[0, 1]$.

There exist two main approaches in MIL:

1. *The instance-level approach*: using a logistic classifier on each instance, then aggregating instance predictions over a bag (e.g. max-pooling, top k-pooling).

2. *The embedding-level approach*: aggregating the instance embeddings, then obtaining a bag-level prediction via a bag-level classifier (e.g. attention-based aggregator, Transformer).

Table 9: Comparison of instance-level performance for the models in Table 3, using a linear classifier trained on the frozen features produced by each model. In addition, we produce bag-level predictions using the maximum output of the linear classifier for each bag.

$(\times 10^{-2})$	SimCLR	Ground-truth	CE finetuning		ItS2CLR			
Instance-level	(CSSL)	finetuning	*w/o iter.*	*iter.*	*w/o both*	*w/o iter.*	*w/o SPL*	*Full*
AUC	96.13	97.56	96.94	96.88	96.64	97.25	96.92	**97.27**
F1-score	85.29	87.69	87.34	86.94	86.00	87.07	87.6	**87.92**
AUPRC	82.65	85.94	79.96	78.02	78.17	**84.56**	77.90	82.09
Dice	49.56	61.39	55.40	54.66	51.85	54.87	55.11	**60.13**
Bag-level (max-pooling)								
AUC	86.25	97.37	87.53	90.54	89.97	93.09	92.81	**97.47**

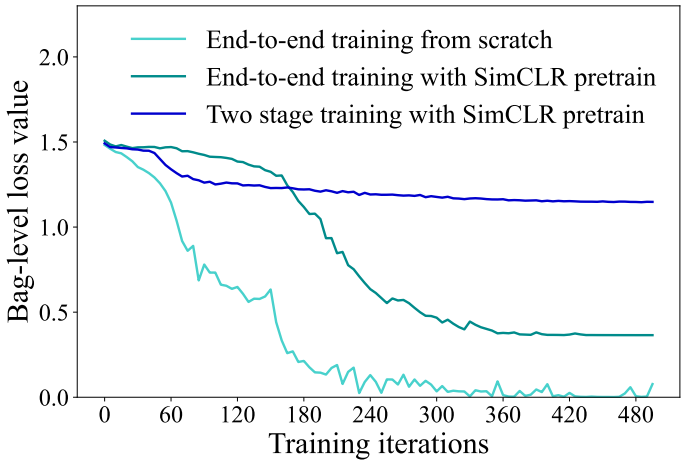

Figure 10: Comparison between end-to-end training and two-stage training on the downsampled version of the Camelyon16 dataset. End-to-end models overfit rapidly.

We denote the embeddings of the instances within a bag by $H = \{h_k\}_{k=1}^{K}$, where K is the number of instances.

Max-pooling obtains bag-level predictions by taking the maximum of the instance-level predictions produced by a logistic instance classifier ϕ, that is

$$g_\phi(H) = \max_{k=1,\cdots,K} \{\phi(h_k)\}\,. \tag{6}$$

Top-k pooling (Shen et al., 2021b) produces bag-level predictios using the mean of the top-M ranked instance-level predictions produced by a logistic instance classifier ϕ, where M is a hyperparameter.

Let top$M$ (ϕ, H) denote the indices of the elements in H for which ϕ produces the highest M scores,

$$g_\phi(H) = \frac{1}{M} \sum_{k \in \text{top}M(\phi,H)} \phi(h_k). \tag{7}$$

Attention-based MIL (Ilse et al., 2018) aggregates instance embeddings using a sum weighted by attention weights. Then

Table 10: Results on the downsampled version of the Camelyon16 dataset.

	End-to-end (scratch)	End-to-end (SimCLR)	SimCLR + DS-MIL	ItS2CLR
Bag AUC	64.52	66.71	80.96	**88.65**
Instance AUC	78.32	81.29	93.94	**95.58**
Instance F-score	51.02	55.71	85.93	**87.01**

Table 11: Results on the Breast Ultrasound dataset.

	SimCLR + Aggregator	End-to-end MIL	ItS2CLR
Bag AUC	80.79	91.26	**93.93**
Bag AUPRC	34.63	58.73	**70.30**
Instance AUC	62.83	82.11	**88.63**
Instance AUPRC	10.58	31.31	**43.71**

the bag-level estimation is computed from the aggregated embeddings by a logistic bag-level classifier φ:

$$g_{\varphi}(H) = \varphi\left(\sum_{k=1}^{K} a_k h_k\right), \tag{8}$$

where a_k is the attention weight on instance k. a_k is computed by:

$$a_k = \frac{\exp\left(w^T \tanh(V h_k^T)\right)}{\sum_{j=1}^{K} \exp\left(w^T \tanh(V h_j^T)\right)}, \tag{9}$$

where $w \in \mathbb{R}^{p\times 1}$ and $V \in \mathbb{R}^{p\times d}$ are learnable parameters and p is the dimension of the hidden layer.

DS-MIL combines instance-level and embedding-level aggregation, we refer to DS-MIL (Li et al., 2021) for more details on this approach.

Transformer (Chen & Krishnan, 2022) aggregation uses an L-layer Transformer to process the set of instance features H. The initial set $H^{(0)}$ is set equal to H. Then it goes through Transformer as following:

$$\begin{aligned} H'^{(l)} &= \text{MSA}\left(H^{(l-1)}\right) + H^{(l-1))} \\ H^{(l)} &= \text{MLP}\left(H'^{(l-1)}\right) + H'^{(l-1)} \end{aligned} \tag{10}$$

for $l = 1, \cdots, L$, where MSA is multiple-head self-attention, MLP is a multi-layer perceptron network. Then the processed vectors $H^{(l)}$ are fed to **Attention-based MIL** (Ilse et al., 2018) to obtain bag-level predictions.

$$g_{\varphi}(H^l) = \varphi\left(\sum_{k=1}^{K} a_k h_k^l\right), \tag{11}$$

where a_k is the attention weight on instance k. a_k is computed by:

$$a_k = \frac{\exp\left(w^T \tanh(V(h_k^l)^T)\right)}{\sum_{j=1}^{K} \exp\left(w^T \tanh(V(h_j^l)^T)\right)}, \tag{12}$$

where $w \in \mathbb{R}^{p\times 1}$ and $V \in \mathbb{R}^{p\times d}$ are learnable parameters and p is the dimension of the hidden layer.

C.2. Implementation Details

Top-k pooling we select the ratio in Top-k pooling from the set $\{0.1\%, 1\%, 3\%, 10\%, 20\%\}$.

DS-MIL The weight between the two cross-entropy loss functions in DS-MIL is selected from the interval $[0.1, 5]$ based on the best validation performance.

Attention-based MIL. The hidden dimension of the attention module to compute the attention weight is set equal to the dimension of the input feature vector (512).

Transformer. We add a light-weighted two-layer Transformer blocks to process instance features. We did not observe increase in performance with additional blocks.

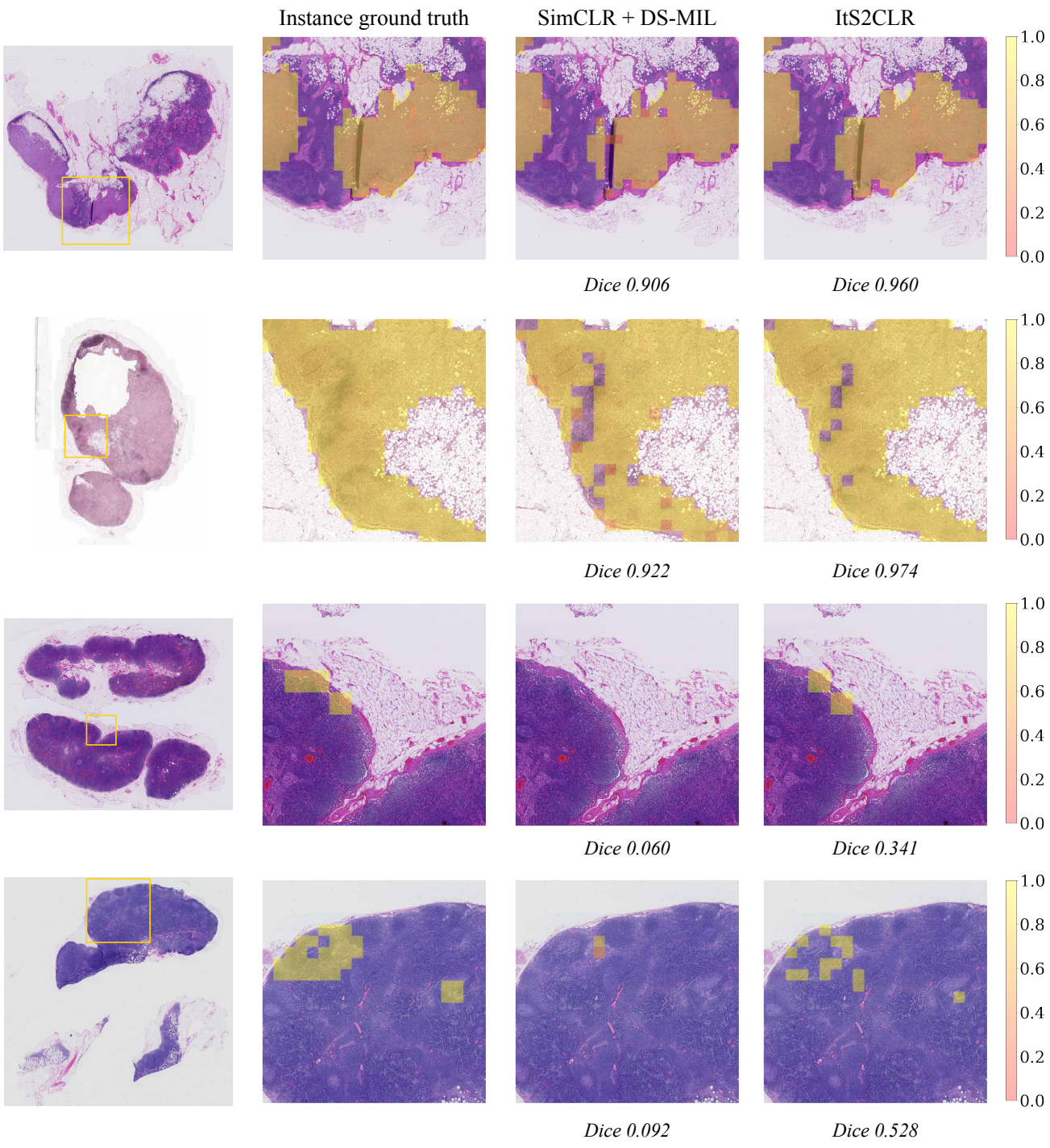

Figure 11: Additional tumor localization maps for histopathology slides from the Camelyon16 test set. Instance predictions are generated by the instance classifier of the DS-MIL trained on extracted instance-level features.