MSc Artificial Intelligence
Master Thesis

Pseudo-label Guided Joint Point, Region and Image-level Contrastive Learning for Task-specific Pretraining

by
Benjamin Conrad
13197401

July 29, 2022

48EC
November 2021 - July 2022

Supervisor:
A Panteli

Examiner:
Dr E Gavves

Universiteit van Amsterdam

Contents

Abstract

Self-supervised contrastive learning is a family of pretraining approaches to learn robust image representations from unlabeled data. While these approaches have recently been able to match the performance of supervised pretraining on natural images, multiple issues have been raised about the effectiveness of these approaches in different image domains. In this work, we will focus on two of these issues. (1) Most contrastive approaches learn image representations that are biased to downstream classification tasks and not optimal for dense prediction tasks. (2) The heuristics used to form similar and dissimilar image pairs do not generalize well outside of single-centric-object images. To address these issues, we propose a new pretraining framework called Point, Region, Image-level Contrast (PRICon). PRICon optimizes a joint point, region and image-level contrastive loss to learn dense image representations which transfers to strong classification and localization performance. To effectively optimize this loss, PRICon incorporates a trained segmentation network to generate pseudo-label masks for the unlabeled data and uses the pseudo-labels to guide the formation of a diverse and accurate set of contrastive pairs. These things combined allow PRICon to learn task-specific image representations in domains where current self-supervised approaches struggle. To demonstrate its effectiveness, we apply PRICon to the domain of histopathology by pretraining on unlabeled whole slide images and fine-tune the pretrained models on various nucleus segmentation benchmarks. We find that PRICon learns image representations from histopathology data with stronger transfer performance than prior approaches and that the pretrained networks are significantly more data efficient when fine-tuning on small amounts of labeled data. We also find that PRICon pretraining is robust to the quality of the pseudo-labels, making the task-specific approach feasible in domains where acquiring a strong labeler network is not possible.

Chapter 1

Introduction

Pretraining via ImageNet [21] classification has long been the standard paradigm to achieve state-of-the-art performance on most computer vision tasks [51]. Supervised pretraining, however, is limited by the amount of label data that can be acquired and, with the introduction of new data-hungry architectures, it has become infeasible to keep up with the demand for magnitudes of more training data [86, 69]. Because of this problem, a shift from supervised to unsupervised/self-supervised pretraining has emerged to learn robust image representations from large collections of unlabeled data [46]. Not only has this shift allowed the continued scaling of model capacities [57, 29], it has also made it possible to perform in-domain pretraining in specialized fields, such as medical imaging, where a large visual discrepancy with natural images exists and acquiring in-domain labeled data is expensive [3, 2].

One of the most commonly adopted classes of self-supervised pretraining approaches is multi-view contrastive learning which learns augmentation invariant image representations by maximizing the similarity between augmented views of the same image and minimizing the similarity between views of different images [15]. While these approaches have been very successful, many works have pointed out that the commonly used image-level discrimination task learns representations that are biased towards classification downstream tasks and are not optimal for dense prediction tasks like object detection and segmentation [76, 41, 53, 6]. This has led to a new class of approaches, called dense contrastive learning, where a contrastive objective is applied at the point or region-level of an image to learn object-centric features. Dense contrastive approaches have been shown to improve the performance in many dense prediction tasks [41, 6]. However, they still inherit a common issue among all self-supervised contrastive approaches, which is, the heuristics used to select similar and dissimilar examples produce many inaccurate contrastive pairs. When training on a dataset of single-centric-object images (e.g. ImageNet [21]) these matching heuristics are generally acceptable, however, when training on images with a large number of object instances from multiple classes (e.g. histopathology and aerial imaging) it becomes difficult to rationalize the use of such heuristics.

In this work, we aim to utilize the strong learning capabilities of dense contrastive approaches and improve the process of forming contrastive feature pairs to enable effective pretraining on a wider range of image domains. To this, we introduce **Point Region Image-level Contrast (PRICon)**, a pretraining framework which simultaneously optimizes point, region and image-level contrastive loss functions with the guidance of pseudo-label masks to generate accurate positive and negative contrastive pairs. By optimizing image representations at multiple levels of granularity, PRICon's unified loss function combines the benefits of standard image-level contrastive approaches and dense contrastive approaches to learn robust object-centric representations that have strong classification and localization capabilities. To make the point and region-level objectives more effective on complex images with many object instances, we incorporate concepts from self-training [80] and use a trained segmentation model to pseudo-label a

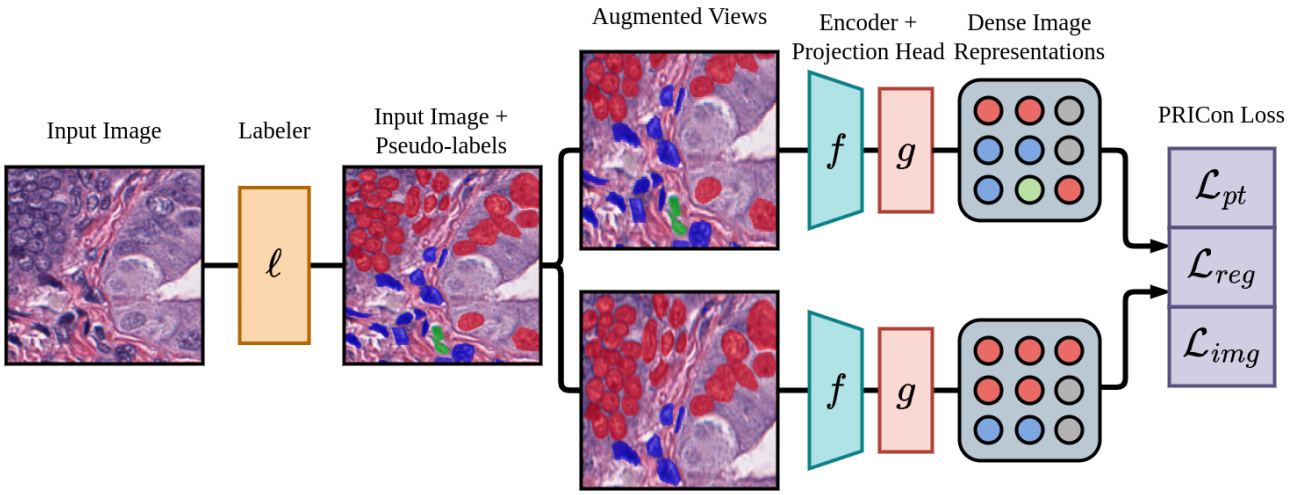


Figure 1.1: Illustration of the Point Region Image-level Contrast (PRICon) pretraining framework. We pass an unlabeled image through a trained labeler network (ℓ) to get a pseudo-label segmentation mask. We then generate two augmented views of the image and pass them through an encoder (f) and projection head (g) to get dense image representations. Using the pseudo-label masks, we assign class labels to each feature vector and jointly optimize point ($\mathcal{L}_{pt}$), region ($\mathcal{L}_{reg}$) and image-level ($\mathcal{L}_{img}$) contrastive loss functions using the pseudo-labels to guide the formation of positive and negative pairs.

large collection of unlabeled data and use the generated segmentation masks to form positive and negative contrastive pairs based on the assigned classes. With the help of the pseudo-labels, PRICon is capable of generating intra-image and inter-image contrastive pairs, it can be trained using a bootstrapping strategy where pseudo-labels are improved during training and, most notably, it has the ability to learn task-specific image representations based on the choice of pseudo-label classes.

To demonstrate the benefits of the task-specific pretraining framework, we apply PRICon to the domain of histopathology by pretraining on unlabeled whole slide images from TCGA [34] and fine-tuning on various nucleus segmentation datasets. Through our experiments, we show that PRICon is more effective at learning from histopathology images than prior self-supervised and semi-supervised approaches, being competitive with ImageNet supervised pretraining, and is significantly more data-efficient when fine-tuning on small amounts of labeled data. Furthermore, we perform an extensive analysis of PRICon to find that the approach is robust to the quality of the pseudo-labels and that the learned point similarities can coarsely segment images without ever learning from true labels.

In summary, the contributions of this work are as follows:

- We propose a unified point, region and image-level contrastive loss function that enables models to learn multiple granularities of image features that transfer well to downstream dense prediction tasks.
- To produce a more diverse and accurate set of contrastive pairs, we utilize a trained pseudo-labeler network to generate segmentation masks for unlabeled data and match points and regions within images based on their pseudo-labeled classes.
- We apply PRICon in the domain of histopathology by pretraining on unlabeled whole slide images from TCGA and find that PRICon is more effective at learning from histopathology images and is significantly more data efficient than prior approaches on multiple nucleus segmentation benchmarks.

Code for this work is available at `https://github.com/bwconrad/pricon`.

Chapter 2

Related Work

2.1 Self-supervised Pretraining

Self-supervised pretraining has been extensively studied as a way to train neural networks without the use of human-annotated data. In natural language processing, self-supervised pretraining, in the form of language modelling, followed by supervised fine-tuning has become the dominant strategy to achieve state-of-the-art performance on nearly every task [22, 10]. The ability to pretrain on massive collections of unlabeled data has allowed models to continue to scale in size and learn surprising emergent properties [47, 10, 19].

In computer vision, many different self-supervised pretext tasks have been proposed in the hopes of finding similar success such as image denoising [73], colorization [87], inpainting [64], jigsaw puzzles [61] and relative position prediction [23]. All of these approaches, however, do not match the performance of the standard supervised classification pretraining [30]. Recently, contrastive learning has emerged as a class of self-supervised methods which can match, and in some cases outperform, supervised pretraining and has been shown to scale better with model capacity and data size [18, 29].

2.1.1 Contrastive Pretraining

Hadsell et al. [38] first proposed learning image representations using a contrastive framework. The framework consists of training a model to produce image representations that are similar between pairs of positive images and dissimilar between pairs of negative images. In modern self-supervised multi-view approaches, positive pairs are defined as two different augmented views of the same image and negative pairs are views from different images [15, 16]. Most works optimize loss functions based on the noise contrast estimation technique [35], however, these losses have the issue of requiring a large number of negative samples in order for the estimator to be accurate [36]. MoCo proposes solving this by using a memory bank to store negative samples which are used when calculating the loss [39]. SimCLR instead takes all in-batch samples as negatives and shows improved performance as the batch size increases [15, 16]. Many works have since improved on the multi-view contrastive framework by dealing with other issues with the approach such as debiasing [20] and decoupling [83] the loss function, removing false negatives [44], generating better negative samples [28] and selecting optimal data augmentations [71]. While contrastive approaches have continued to improve over time, optimal performance still requires a large number of negative samples, making contrastive pretraining computationally expensive.

An alternate approach which circumvents issues with noise contrast estimation-based losses, is to do contrastive learning without any negative samples. First introduced in BYOL [33], positive-only or non-contrastive methods, learns image representations by only enforcing that

positive pairs are similar to each other. In BYOL [33], two augmented views are passed through an online and a target network, respectively, with the objective of the online network learning to predict the target network's image representation. A key challenge with non-contrastive approaches is preventing the model from collapsing to a loss of zero by outputting the same features for all images. BYOL prevents this by updating the target network as the exponential moving average of the online network [33] while other methods propose solutions such as stop gradients [17], feature whitening [24], feature decorrelation [43] and feature centering [11]. Many similarity functions have also been explored for the loss such as mean squared error [33, 17], cross-entropy [11], cross-correlation [85], covariance [7] and KL-divergence [74]. Non-contrastive approaches achieve similar performance to standard contrastive methods with the added benefit of being less influenced by the batch size [33, 85, 7], however, there is still little theoretical understanding on what prevents non-contrastive methods from collapsing which can make training unstable [72, 75].

2.1.2 Dense Contrastive Pretraining

Most self-supervised contrastive approaches are designed to improve downstream classification tasks by representing images as a single feature vector which ignores any spatial information and assumes the existence of only one object class per image. To fix this, some works have looked into modifying the standard contrastive framework to learn image representations that are better tailored for dense prediction tasks such as object detection and segmentation [76, 53, 56, 6, 41, 81, 78, 77, 13]. Instead of applying a contrastive loss on globally pooled feature vectors, the key idea of dense approaches is to instead take the non-pooled dense feature maps and use a matching heuristic to form pairs of feature vectors to which the contrastive loss is applied to. In point-level approaches, individual feature vectors are sampled from the dense representations and positive and negative pairs are formed based on heuristics such as the similarity score [76], weighted spatial distance [81], earths mover's distance [56] and predefined image gridding [6]. In region-level approaches, feature vectors are generated by pooling regions with similar semantics within an image to perform improved object-centric contrasting. Different region proposal techniques have been proposed such as unsupervised super-pixel algorithms [41], sliding windows [78], selective search [77] and clustering [13]. An additional benefit with most dense approaches is the ability to perform both inter-image and intra-image contrast, however, most matching heuristics only apply within the same image, limiting the number of inter-image pairs.

In PRICon, we take the benefits of standard and dense contrastive approaches and jointly learn point, region and image-level contrastive objectives to capture both global and local image features which are essential for dense prediction tasks.

2.2 Semi-supervised Learning

Semi-supervised learning [88] is a paradigm of learning approaches to train on both labeled and unlabeled data. Closely related to contrastive methods, consistency training [70, 4] enforces models to be invariant to noise injected into the inputs or model parameters. In the commonly adopted Mean Teacher approach [70, 65], a student network is trained by optimizing a supervised loss function, such as cross-entropy, on labeled data and a consistency loss on unlabeled data that has been pseudo-labeled by a teacher network whose weights are the exponential moving average of the student's. Originally, basic image augmentations such as random affine transformations and Gaussian noise are applied to input images [70], however, later works have found that stronger augmentations such as CutMix [25] and ClassMix [62] yield improved

empirical results. Regional Contrast (ReCo) [55] extends the mean teacher framework by introducing concepts from dense contrastive approaches, namely, a pixel-level contrastive objective. A number of positive and negative point pairs are sampled from the student's feature maps based on the teacher's pseudo-labels and an intra-image contrastive loss is optimized alongside the supervised and consistency losses.

2.2.1 Self-training

An issue with semi-supervised consistency training approaches is that since the student and teacher networks are trained simultaneously from scratch, the pseudo-labels early in training are very noisy, leading to a poor learning signal for the consistency loss. To alleviate this problem, self-training uses a frozen, already trained teacher network to generate the unlabeled data's pseudo-labels [80, 89, 16]. Self-training shares similarities with knowledge distillation [42] by allowing the student and teacher networks to have different architectures unlike with standard semi-supervised approaches [16]. Chen et al. [16] proposed combining pretraining and self-training by training a large task-agnostic model on unlabeled data, fine-tuning the model on labeled data and end with training a small task-specific model via self-training using the large model as the pseudo-labeler.

PRICon follows a similar strategy of using a fully trained segmentation model to generate pseudo-labels for the unlabeled data. However instead of training on both labeled and unlabeled data simultaneously, we separate training into pretraining on unlabeled data and fine-tuning on labeled data which yields better empirical performance.

2.3 Pretraining for Medical Imaging

Despite the large domain shift, pretraining on natural image datasets, like ImageNet [21], has become the default approach to achieve state-of-the-art results on most medical imaging tasks [32, 1, 12, 79]. While some works have achieved improved results with pretraining on labeled in-domain data [14, 54], acquiring large quantities of annotated data is generally infeasible in the field because of the requirement of medical specialists.

To remove the need for labeled data, work has been done in applying self-supervised pretraining approaches to the medical domain. Multiple works train general self-supervised approaches on in-domain datasets [68, 2, 67], while others modify these approaches to better tailor medical data such as adding new data augmentations [82], using patient information [3] and incorporating relative patch position [52]. Other works have proposed new self-supervised tasks specifically for medical data such as patch outpainting [9], anatomical position prediction [5] and zoom magnification prediction [50]. An issue with most of these pretraining studies is that they focus on classification downstream tasks and do not consider transferring to dense prediction tasks.

Chapter 3

Methodology

In this section, we introduce Point Region Image-level Contrast (PRICon), a contrastive pre-training framework which learns image representations using a joint point, region and image-level contrastive objective under the guidance of pseudo-labels. We will start by covering the general multi-view contrastive learning framework used in SimCLR [15, 16] which PRICon follows. We then introduce our unified point, region and image-level contrastive loss by detailing how we generate pseudo-label segmentation masks used in the point and region-level losses and describing the formulation for each of the three objectives.

3.1 Contrastive Framework

PRICon builds off the image-level multi-view contrastive learning framework used in SimCLR [15, 16]. SimCLR learns image representations/features in a self-supervised manner by maximizing the similarity between representations from different augmented views of the same image and minimizing the similarity between representations of different images.

The SimCLR framework is described as follows. Given an unlabeled image $\mathbf{x}_k \in \mathbb{R}^{H \times W \times 3}$, two sets of random augmentations are applied to produce image views $\mathbf{v}_i$ and $\mathbf{v}_j$. These augmentations include random crop and resize, color jitter and Gaussian blur, among others. The two views are passed through an encoder network $f(\cdot)$ (e.g. a ResNet [40]) to generate representations $\mathbf{h}_i$ and $\mathbf{h}_j$ and then through an MLP projection head $g(\cdot)$ to get $\mathbf{z}_i$ and $\mathbf{z}_j$ which are used for the contrastive loss. Since SimCLR is an image-level approach, $\mathbf{h}_i$ and $\mathbf{h}_j$ are extracted from after the encoder's global pooling layer to aggregate the image representations into D-dimensional feature vectors.

To perform contrastive learning, SimCLR defines an instance discrimination task among the B images within a mini-batch. The goal of the task is to train the encoder to generate similar representations between positive image pairs and dissimilar representations between negative pairs. Here, positive image pairs are formed between the augmented views of the same image ($\mathbf{v}_i$ and $\mathbf{v}_j$) and negative pairs with the other $2(B-1)$ views. The encoder is trained on this instance discrimination task by optimizing, for a pair of positive examples (i, j), the following contrastive loss function:

$$\mathcal{L}_{img}^{i,j} = -\log \frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_j)/\tau)}{\sum_{k=1}^{2B} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_k)/\tau)} \tag{3.1}$$

where $\text{sim}(\cdot, \cdot)$ is the cosine similarity and τ is a temperature hyperparameter [15]. The loss over is averaged over all positive pairs (i, j) and (j, i) within a mini-batch.

3.2 Point, Region and Image-level Contrast

Multiple works have pointed out issues with image-level contrastive learning approaches [33, 20, 83, 76, 44]. In this work, we will focus on two of these issues. **First**, image-level contrastive approaches learn representations that are scale and translation invariant, making the pretraining biased towards downstream classification tasks [76]. In tasks such as object detection and segmentation, these invariances are undesirable since the position and size of objects are required for localization. **Second**, a significant number of false-positive and false-negative pairs are generated because of the heuristics used to form contrastive pairs in self-supervised approaches. This causes the learning signal to be noisy and is amplified when training on complex images with multiple object instances from different classes.

We address the first issue by modifying the SimCLR framework to learn dense feature representations by simultaneously optimizing point, region and image-level contrastive objectives. For the second issue, we utilize a trained segmentation network to generate pseudo-label segmentation masks for unlabeled data and use the masks to form more accurate positive and negative image pairs in the point and region-level losses. All of these together form our PRICon pretraining framework which we illustrate in Figure 1.1.

3.2.1 Pseudo-label Generation

The core idea of contrastive learning methods is to form positive and negative pairs of image representations and learn an encoder that makes the representations of positive pairs similar to each other and negative pairs dissimilar. For image-level approaches like SimCLR [15, 16], image representations are encoded as a single feature vector, however, in dense contrastive approaches [76, 81, 56, 41, 6] the representations are non-pooled dense feature maps and contrastive learning is applied between different spatial regions within an image. In both cases, learning is generally done in a self-supervised manner using different heuristics to generate the positive and negative pairs [15, 76, 41, 6]. However, these heuristics often generate many false-positive and false-negative pairs. To alleviate this issue, we opt to take ideas from self-training [80] and use a trained segmentation network to pseudo-label an unlabeled dataset and generate segmentation masks which we can use to form more accurate contrastive pairs.

For each unlabeled image $\mathbf{x}_k \in \mathbb{R}^{H\times W\times 3}$, we use a labeler network $\ell(\cdot)$ to generate a corresponding multi-class semantic segmentation mask $\mathbf{m}_k \in \mathbb{R}^{H\times W\times C}$, where $C \geq 2$ is the number of classes and each class channel is a binary mask. We assign a single label to each pixel, therefore, when $C = 2$ the labeler assigns each pixel as either foreground or background and when $C > 2$ a pixel is either background or one of $C - 1$ object classes.

We train the labeler beforehand on a fully annotated segmentation dataset in the same domain as the unlabeled pretraining dataset (e.g. natural or medical images). In our experiments, we use the same segmentation network architecture for the labeler and during fine-tuning, but there are no requirements on the labeler's architecture. This allows the architectures of the two to differ similar to in knowledge distillation frameworks [42, 16]. In this work, we focus on a bootstrapping approach where we train the labeler on the same dataset as the target fine-tuning task, however, in general the datasets and object classes can differ. The choice of pseudo-label classes, however, largely determines the number of modes that will emerge within the learned representation space and is the key factor that makes PRICon pretraining task-specific.

3.2.2 Multi-level Contrastive Framework

PRICon builds off the SimCLR framework and learns dense image representations by following a unified point, region and image-level objective. The general framework of PRICon is as follows. Given an unlabeled image $\mathbf{x}_k$ and corresponding pseudo-label mask $\mathbf{m}_k$, we create two

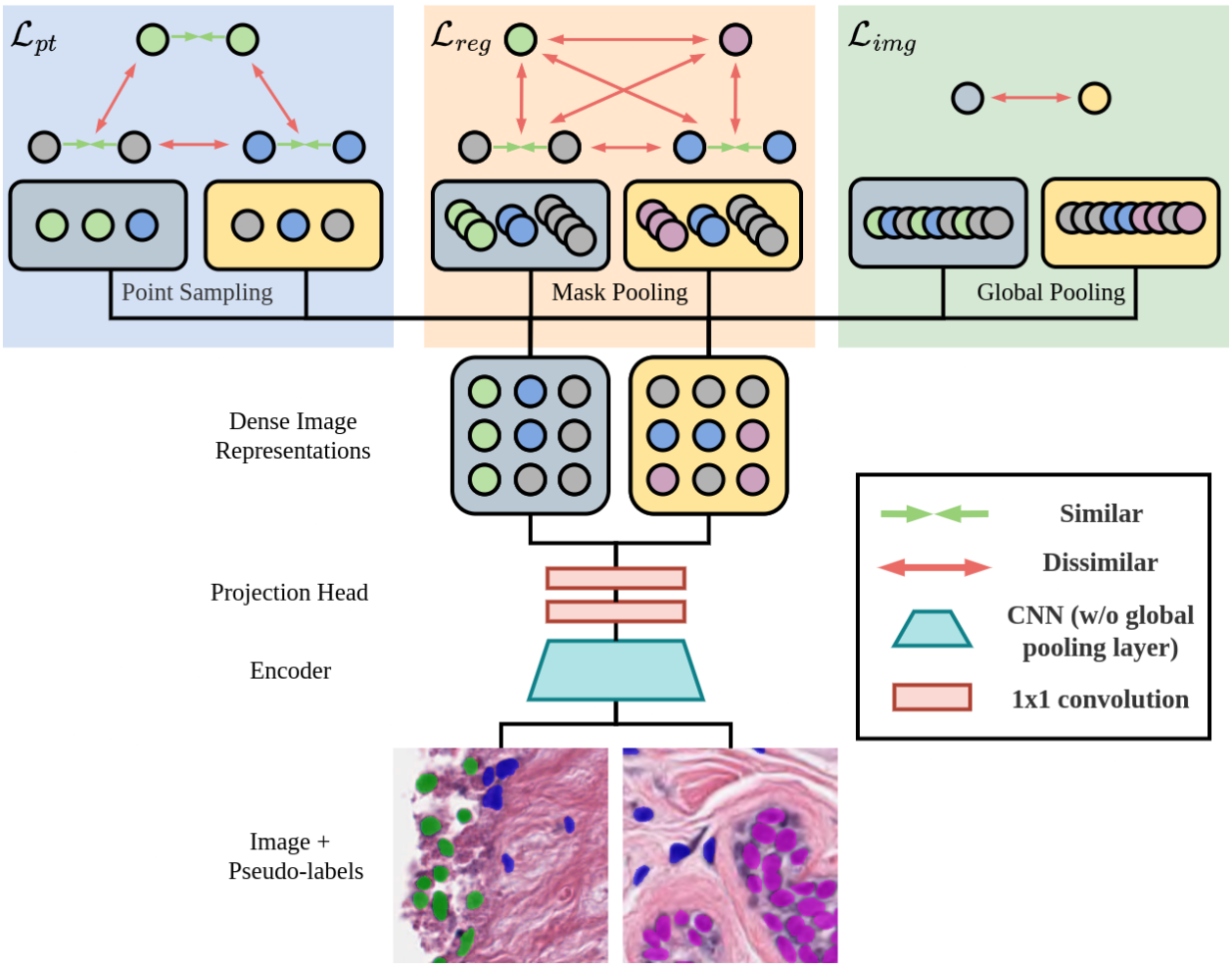


Figure 3.1: Illustration of the point ($\mathcal{L}_{pt}$), region ($\mathcal{L}_{reg}$) and image-level ($\mathcal{L}_{img}$) loss functions. We pass images through an encoder and projection head to extract dense image representations and assign class labels to the feature vectors based off the pseudo-label masks. Each of the losses differs in how we sample and/or aggregate the dense representations. **Point-level**, we sample N point feature vectors from the dense feature maps of each image view. **Region-level**, we apply mask pooling to aggregate all feature vectors of the same class within an image view into per-class region features. **Image-level**, we apply global pooling on all feature vectors to generate per-image features. A contrastive loss is applied to the representations based on their class pseudo-label for the point and region-level losses and on whether the image view comes from the same image for the image-level loss.

augmented views $\mathbf{x}_i$ and $\mathbf{x}_j$ and additionally generate corresponding augmented masks $\mathbf{m}_i$ and $\mathbf{m}_j$ by applying the same geometric augmentations (i.e. crop, flip, rotation) to $\mathbf{m}_k$. We pass the views through an encoder network $f(\cdot)$, however different from image-level approaches, we extract the dense feature maps before the global pooling layer such that $\mathbf{h}_i, \mathbf{h}_j \in \mathbb{R}^{R \times R \times D}$. This means, when using a ResNet-18 [40] encoder and 224×224 resolution input images, we extract representations of size $7 \times 7 \times 512$. We then pass $\mathbf{h}_i$ and $\mathbf{h}_j$ through a convolutional projection head consisting of 1×1 convolutions to get $\mathbf{z}_i$ and $\mathbf{z}_j$ which maintain the same $R \times R$ spatial resolution. For the masks, we downsample $\mathbf{m}_i$ and $\mathbf{m}_j$ to $R \times R$ resolution using average pooling. Using the dense representations and corresponding masks, we simultaneously optimize three contrastive loss functions which we illustrate in Figure 3.1.

Point-level Loss

With our multi-class segmentation masks $\mathbf{m}_i$ and $\mathbf{m}_j$, we randomly select a single class mask $\mathbf{m}_i^c, \mathbf{m}_j^c \in \mathbb{R}^{R \times R}$ and normalize the masks into multinomial probability distributions. We then sample from each mask a single point (u_i, v_i) and (u_j, v_j) and extract the corresponding point features by indexing the dense feature map at those points:

$$\mathbf{p}_i = \mathbf{z}_i[u_i, v_i] \qquad \mathbf{p}_j = \mathbf{z}_j[u_j, v_j]$$

The process of randomly selecting a class and sampling point features is repeated N times on each view in the mini-batch to form a set of point representations $\mathbf{p}$ of size $|\mathbf{p}| = 2BN$. While doing this, we also track the corresponding class of each point feature into a class vector $\mathbf{a}^p$. Since the classes in the masks are the same across all images (unlike when using an unsupervised approach [41]), we generate both intra-image and inter-image positive and negative point pairs between all feature vectors that share or do not share the same pseudo-label class. With the set of point pairs, we optimize a point-level contrastive loss which we define for a pair of positive points (i, j) as:

$$\mathcal{L}_{pt}^{i,j} = -\log \frac{\exp(\text{sim}(\mathbf{p}_i, \mathbf{p}_j)/\tau)}{\sum_{k=1}^{|\mathbf{p}|} \mathbb{1}_{[a_k^p \neq a_i^p]} \exp(\text{sim}(\mathbf{p}_i, \mathbf{p}_k)/\tau)} \tag{3.2}$$

where $\text{sim}(\cdot, \cdot)$ is the cosine similarity and τ is a temperature hyperparameter.

By contrasting only single points within dense image representations, the encoder learns better low-level features by limiting the amount of context that is available to discriminate with. In Section 4.3, we show that this helps the encoder distinguish object boarders more accurately compared to image-level approaches.

Region-level Loss

For each class $c \in \{1 \dots C\}$, we take the binary class masks $\mathbf{m}_i^c$ and $\mathbf{m}_j^c$ and generate aggregated class regions feature vectors using mask pooling [41]:

$$\mathbf{h}_k = \frac{1}{\sum \mathbf{m}_i^c} \sum_{u,v} \mathbf{m}_i^c[u, v] \cdot \mathbf{z}_i[u, v] \tag{3.3}$$

When a mask is empty, we do not calculate a feature vector for that class, therefore, the set of region features $\mathbf{h}$ is of size $|\mathbf{h}| \leq 2BC$. Again, we track the class of each region vector in a class vector $\mathbf{a}^r$ and use it to generate intra and inter-image region feature pairs based on their pseudo-label classes. With the set of region pairs, we optimize a region-level contrastive loss which we define for a pair of positive regions (i, j) as:

$$\mathcal{L}_{reg}^{i,j} = -\log \frac{\exp(\text{sim}(h_i, h_j)/\tau)}{\sum_{k=1}^{|h|} \mathbb{1}_{[a_k^r \neq a_i^r]} \exp(\text{sim}(h_i, h_k)/\tau)} \tag{3.4}$$

Region-level contrast provides a more holistic view of object instances compared to the point-level objective and enforces representations to be consistent over the entirety of an object class type. Along with the point-level loss, these objectives are more tailored for learning from images that contain many object instances and they can be effectively optimized with the guidance of the pseudo-labels. The region-level loss closely follows the loss used in DetCon [41].

Image-level Loss

For image-level contrast, we use the same loss used in SimCLR, outlined in Equation 3.1, and do not use the pseudo-label masks within it. Since $\mathbf{z}_i$ and $\mathbf{z}_j$ are dense representations, we apply global average pooling to generate a single feature vector for each image view. Applying pooling after the projection head instead of before slightly differs our image-level loss from SimCLR's, however, this change allows our framework to only require a single convolutional projection head which saves computation cost. Since dense prediction tasks can be decomposed into classification and localization sub-tasks, including an image-level loss helps improve classification performance as shown by prior dense contrastive approaches [76, 6, 53].

Overall Loss

In the overall loss function, we jointly optimize a weighted sum of the point, region and image-level losses:

$$\mathcal{L} = \alpha_{pt}\mathcal{L}_{pt} + \alpha_{reg}\mathcal{L}_{reg} + \alpha_{img}\mathcal{L}_{img} \tag{3.5}$$

where α_{pt}, α_{reg} and α_{img} are the corresponding loss weight hyperparameters which we set to values ≥ 0.

3.2.3 Cyclic Training

For this work, we focus on applying PRICon in the setting where the labeler and fine-tuning models share the same segmentation network architecture and training dataset. In this setting, a natural extension to the pretraining followed by fine-tuning training strategy is to use bootstrapping to repeatedly update the labeler with the newly fine-tuned model and continue pretraining using the updated pseudo-labels.

Specifically, our cyclic training strategy is the repeated application of the following process. We first train a labeler network from scratch on the target fine-tuning dataset and use the labeler to pretrain an encoder with PRICon for E epochs. Next, we initialize a segmentation network with the pretrained encoder and fine-tune the network on the target dataset. After fine-tuning, we then replace the labeler with the newly trained segmentation network and continue pretraining the same encoder for another E epochs using pseudo-labels from the updated labeler.

Since cyclic training makes the point and region-level objectives stricter as the labeler improves during training, we can view the training strategy as a form of curriculum learning [37]. A benefit of this strategy, and curriculum learning frameworks in general, is that as the training objective changes over time, the loss landscape similarly transforms along with it which can help models escape local minima and continue learning for longer [37].

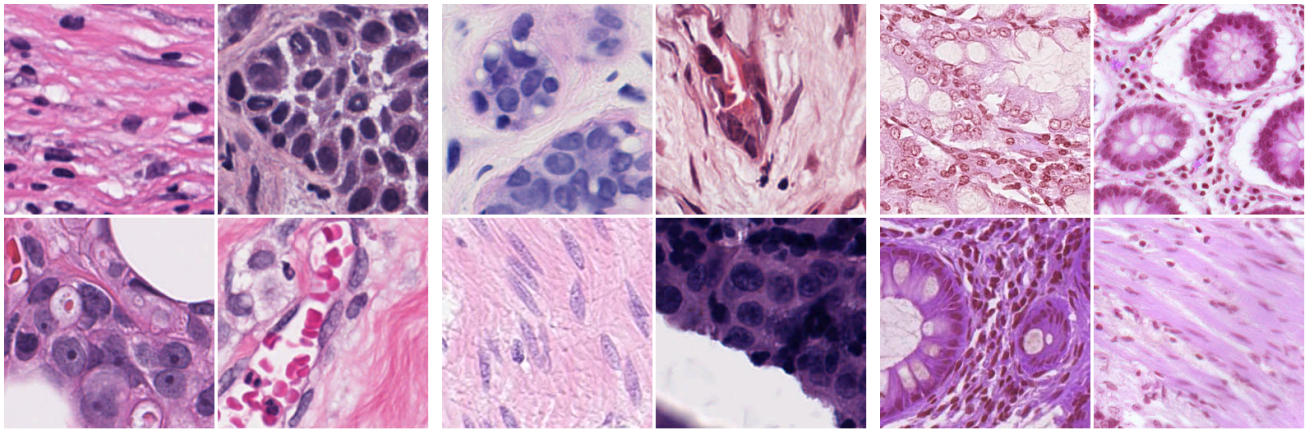
Chapter 4

Experiments

In this section, we explore PRICon's effectiveness in learning task-specific image representations for histopathology downstream tasks by pretraining on unlabeled whole slide images and fine-tuning on various nucleus segmentation datasets.

4.1 Experimental Settings

4.1.1 Data



(a) TCGA (b) Pannuke (c) Lizard

Figure 4.1: Example images from the (a) TCGA, (b) Pannuke and (c) Lizard dataset.

TCGA

For pretraining, we use digitized H&E stained whole slide images (WSI) of breast cancer from the Cancer Genome Atlus (TCGA) [34]. We tile the slides into 512×512 non-overlapping patches at a zoom magnification of $40\times$ for a total of 150,000 unlabeled images. To avoid background regions and artifacts, we tile from manually selected regions of interest in the WSIs. We show samples from the dataset in Figure 4.1a.

Pannuke

Pannuke [26, 27] is a multi-class nucleus segmentation dataset that consists of 7,904 patches from H&W stained WSIs of 19 different cancer types. The patches are at a resolution of 256×256 and are taken at $40\times$ magnification. Nuclei are classified into five classes: neoplastic,

non-neoplastic epithelial, inflammatory, connective and dead. All other pixels are labeled as background. The dataset is split into three folds of approximately equal size as proposed by the original authors [27]. We show samples from the dataset in Figure 4.1b.

In our experiments, we also use the subset of only breast cancer patches, which we refer to as Pannuke-Breast. The subset contains 2,351 patches which we split into training, validation and test sets of 1,692, 188 and 471 images, respectively.

Lizard

Lizard [31] is a multi-class nucleus segmentation dataset that consists of 4,981 patches from H&E stained WSIs of colon cancer. The patches are at a resolution of 256×256 and are taken at $20\times$ magnification. Nuclei are classified into six classes: epithelial, lymphocyte, plasma, neutrophil, eosinophil and connective tissue. All other pixels are labeled as background. The dataset is split into three folds of approximately equal size as proposed by the original authors [31]. We show samples from the dataset in Figure 4.1c.

4.1.2 Pretraining Settings

Data Augmentations

PRICon follows a multi-view contrastive framework where we transform each input image with a series of random augmentations to generate two unique image views. For the augmentations, we use random crop and resize to 224×224 resolution, horizontal flip, rotations in increments of $90°$, color jitter and grayscale conversion. We always apply random crop and resize and apply the other augmentations randomly with some probability. More details about the augmentations and their hyperparameter values can be found in Appendix A.1. Notably, we do not use Gaussian blurring as it has been shown to harm contrastive learning performance on histopathology images [68], which we also observed during experimentation.

When we pass images through the labeler, we do not apply any augmentations to ensure optimal mask quality. On the resulting masks, we apply the same geometric augmentations (crop and resize, horizontal flip and rotation) as used in the corresponding image view to align the mask and view.

Architecture

We pretrain ResNet-18 [40] encoders without the global averaging pooling and final classification layer. The projection head consists of two 1×1 convolution layers with a hidden dimension of 2048 and an output dimension of 128 and a ReLU activation between the convolution layers. We use the same projection head for all three objectives. We experimented with task-specific heads, similar to DenseCL [76], but observed no performance gain. The projection head architecture is identical to that used in SimCLR except we replace the linear layers with 1×1 convolutions which makes the PRICon architecture have the same number of parameters as SimCLR. The labeler network is a U-Net [66] segmentation model with a ResNet-18 encoder.

Optimization

We optimize using LARS [84] with a batch size of 256 and weight decay of 10^{-4}. We exclude the bias and batch normalization parameters from the LARS adaptation. The initial learning rate is 0.2 and we decay it to 0 following a cosine annealing schedule [58] over 100 epochs on the TCGA dataset. For the point, region and image-level loss weights, we ran a random search [8] for 20 iterations and found the best results with $\alpha_p = 0.7$, $\alpha_r = 0.75$ and $\alpha_i = 0.15$. For

the point-level loss, we sample $N = 32$ points and for all three losses we use a temperature of $\tau = 0.1$. We pretrain using mixed precision [59] on a single Nvidia RTX 3090 GPU.

We train the labeler network from scratch following the same settings used during fine-tuning which we will describe in Section 4.1.3.

4.1.3 Fine-tuning Settings

To evaluate PRICon, we fine-tune the pretrained encoder on labeled nucleus segmentation datasets. We fine-tune a U-Net [66] segmentation network with a ResNet-18 [40] encoder initialized with the pretrained weights. We use the Adam optimizer [48] with a β_1 of 0.9, β_2 of 0.999, weight decay of 0 and batch size of 16. The initial learning rate is 10^{-3} and we decay to 0 following a cosine annealing schedule [58] for 100 epochs. During training, we apply random horizontal and vertical flip, rotations in increments of 90°, Gaussian blur, Gaussian noise data augmentations (more details in Appendix A.1). During testing, we apply no augmentations. For the evaluation metric, we use multi-class panoptic quality (PQ) [49] as that is the default metric used for the Pannuke dataset [26, 27]. On the Pannuke and Lizard datasets, we report the average cross-validation PQ over the three data folds, training on two of the folds and evaluating on one. On Pannuke-Breast, we report the average PQ on the test set over three independent training runs.

We fine-tune using two different strategies: fine-tuning both the U-Net's encoder and decoder (referred to as *full fine-tuning*) and fine-tuning only the decoder while keeping the pretrained encoder frozen (referred to as *frozen encoder*). The frozen encoder setting is analogous to the linear classification experiments done in many self-supervised works and it is an important analysis tool to measure the quality of the pretrained representations [15, 33, 85, 39].

4.1.4 Baseline Settings

We compare PRICon against four classes of baselines: no pretraining, supervised pretraining, self-supervised pretraining and semi-supervised training. The supervised model is pretrained on ImageNet classification [21]. For the self-supervised baselines, we train encoders with SimCLR [15, 16] and SimSiam [17] on the TCGA dataset. For the semi-supervised baselines, we train CutMix [25], ClassMix [62] and ReCo [55] using the TCGA dataset for unlabeled data and the target fine-tuning dataset for labeled data. Semi-supervised approaches do not adopt a pretraining followed by fine-tuning process but rather only perform a single training run so are only included in full fine-tuning experiments. We provide detailed implementation details for the baselines in Appendix A.2.

4.2 Results

4.2.1 Pannuke Fine-tuning

In Table 4.1 we report fine-tuning results on the Pannuke dataset [26, 27]. For PRICon, we train the labeler from scratch on folds 1 and 2 of Pannuke. When we keep the encoder frozen and fine-tune only the decoder, PRICon outperforms all baselines and surpasses ImageNet supervised pretraining by 4.8% (+0.0174 PQ). Comparing with the semi and self-supervised baselines, which are also trained on TCGA data, we see an even larger improvement which indicates that PRICon's performance is not simply from pretraining on in-domain data but rather the method itself. When we do full fine-tuning, PRICon outperforms all semi-supervised and self-supervised baselines, improving on SimCLR by 0.9% (+0.0035 PQ), showing that it is more effective at learning from unlabeled histopathology data than prior approaches. We

Method	PQ	
	Frozen Encoder	Full Fine-tuning
Scratch	-	0.3773
Supervised (ImageNet)	0.3641	**0.3977**
SimCLR [15]	0.3610	0.3896
SimSiam [17]	0.2973	0.3680
CutMix [25]	-	0.3265
ClassMix [62]	-	0.3331
ReCo [55]	-	0.3257
PRICon	**0.3815**	0.3931

Table 4.1: Results on the Pannuke dataset. We fine-tune a U-Net segmentation network with a pretrained ResNet-18 encoder for 100 epochs and report the average panoptic quality across three data folds. *Frozen encoder* refers to fine-tuning only the decoder and keeping the pretrained encoder frozen. *Full fine-tuning* refers to fine-tuning both the encoder and decoder.

do, however, see PRICon fall to the supervised model by 1.2% (-0.0046 PQ). We found that because of PRICon's task-specific framework, it can learn subtle biases from experimental design choices which can impact its downstream performance. Here we suspect that since our TCGA pretraining dataset only contains breast cancer images and Pannuke contains images of 19 different cancer types, PRICon learns weights that are overly specific to breast cancer, making the encoder struggle to adapt to different cancer types.

Method	PQ	
	Frozen Encoder	Full Fine-tuning
Scratch	-	0.4168
Supervised (ImageNet)	0.3958	0.4252
SimCLR [15]	0.3921	0.4150
PRICon	**0.4019**	**0.4300**

Table 4.2: Results on the Pannuke-Breast dataset. We fine-tune a U-Net segmentation network with a pretrained ResNet-18 encoder for 100 epochs and report the average panoptic quality on the test set over three independent runs. *Frozen encoder* refers to fine-tuning only the decoder and keeping the pretrained encoder frozen. *Full fine-tuning* refers to fine-tuning both the encoder and decoder.

To test this hypothesis, we fine-tune the same encoders on the Pannuke-Breast dataset so that the pretraining and fine-tuning datasets are both of only breast cancer. In Table 4.2 we report the fine-tuning results of these experiments. PRICon achieves the best results, outperforming ImageNet supervised pretraining in both the frozen encoder and full fine-tuning settings by 1.5% (+0.0061 PQ) and 1.2% (+0.0048 PQ) respectively. We find that SimCLR does not have a similar relative improvement, and actually falls behind training from scratch, which demonstrates how PRICon's task-specific pretraining is better at capitalizing on accurately aligning the pretraining data domain with the target task. We speculate that training on a

large diverse dataset of multiple cancer types will help learn more general weights that perform better on the full Pannuke dataset and leave this for future work.

4.2.2 Lizard Fine-tuning

Method	PQ	
	Frozen Encoder	Full Fine-tuning
Scratch	-	0.3420
Supervised (ImageNet)	**0.2888**	**0.3519**
SimCLR [15]	0.2601	0.3381
PRICon	0.2785	0.3474

Table 4.3: Results on the Lizard dataset. We fine-tune a U-Net segmentation network with a pretrained ResNet-18 encoder for 100 epochs and report the average panoptic quality across three data folds. *Frozen encoder* refers to fine-tuning only the decoder and keeping the pretrained encoder frozen. *Full fine-tuning* refers to fine-tuning both the encoder and decoder.

In Table 4.3 we report fine-tuning results on the Lizard dataset [31]. For PRICon, we train the labeler from scratch on folds 1 and 2 of Lizard. Similar to Pannuke, PRICon demonstrates improved pretraining capabilities on TCGA data, outperforming SimCLR by 7.1% (+0.1840 PQ) and 2.8% (+0.0093 PQ) in the frozen-encoder and full fine-tuning settings, respectively. Comparing to supervised pretraining, PRICon falls 3.6% (-0.0103 PQ) in the frozen encoder setting, but only 0.9% (-0.0033 PQ) when doing full fine-tuning similar to Pannuke.

The Lizard dataset has notable differences from the TCGA pretraining dataset. Specifically, Lizard contains colon cancer images taken at 20× magnification while our TCGA dataset contains breast cancer images taken at 40× magnification. We speculate that the difference in magnification is the main cause of the poor frozen encoder performance since the encoder is biased to 40× images and cannot unlearn this bias. This again highlights how PRICon's task-specific pretraining limits its generalization capabilities and that carefully selecting pretraining data is essential to achieve optimal results.

4.2.3 Data Limited Fine-tuning

In the previous experiments, we discussed how PRICon's task-specific pretraining can lead to mixed results in terms of generalization. We next explore a major benefit of task-specific pretraining which is improved data efficiency when fine-tuning on limited labeled data. In these experiments, we fine-tune pretrained models on subsets of 1% or 10% of the Pannuke dataset. For each of the dataset's three folds, we create subsets with 1% and 10% of the original images and train on two of the reduced folds while evaluating on the other fold containing all 100% of its original images. In this protocol, the 1% training sets contain ~ 50 images and the 10% training sets contain ~ 500 images. We follow the same fine-tuning settings as the previous experiments except models are fine-tuned for 200 epochs.

We report the results of the experiments in Table 4.4. PRICon outperforms all other approaches in each of our data limited experiments. When using 1% of the training data, PRICon surpasses supervised pretraining by 37.2% (+0.0512 PQ) and 14.0% (+0.0243 PQ) in the frozen encoder and full fine-tuning settings, respectively. We even see that the PRICon frozen encoder model outperforms all other fully fine-tuned baselines by a substantial margin.

Method	1%		10%	
	Frozen Encoder	Full Fine-tuning	Frozen Encoder	Full Fine-tuning
Scratch	-	0.1404	-	0.2863
Supervised (ImageNet)	0.1375	0.1739	0.2737	0.3048
SimCLR [15]	0.1288	0.1680	0.2627	0.2956
PRICon	**0.1867**	**0.1982**	**0.2924**	**0.3080**

Table 4.4: Results on the Pannuke dataset using 1% and 10% of the training data. We fine-tune a U-Net segmentation network with a pretrained ResNet-18 encoder for 200 epochs and report the average panoptic quality across three data folds. The training data folds consist of either 1% or 10% of their images while the test fold consists of 100% of their images. *Frozen encoder* refers to fine-tuning only the decoder and keeping the pretrained encoder frozen. *Full fine-tuning* refers to fine-tuning both the encoder and decoder.

When using 10% of the training data, PRICon improves 6.8% (+0.1870 PQ) and 1.0% (+0.0032 PQ) over supervised pretraining in the frozen encoder and full fine-tuning settings, respectively. These results show that by learning task-specific features from unlabeled data, PRICon can significantly improve the performance on tasks with very limited labeled training data.

4.2.4 Cyclic Training

Method	TCGA Epochs	PQ	
		Frozen Encoder	Full Fine-tuning
Supervised (ImageNet)	0	0.3641	0.3977
PRICon	100	**0.3815**	0.3931
PRICon (Cyclic)	25	0.3769	0.3895
	50	0.3755	0.3943
	75	0.3811	0.3950
	100	**0.3816**	**0.3990**

Table 4.5: Results on the Pannuke dataset using the cyclic training strategy. During each training iteration, we pretrain for 25 epochs on TCGA, fine-tune on Pannuke for 100 epochs and update the labeler with the newly fine-tuned model. This process is repeated four times for a total of 100 pretraining epochs. *TCGA epochs* is the number of completed pretraining epochs on the TCGA dataset.

In Table 4.5 we report fine-tuning results on the Pannuke dataset following our cyclic training strategy. During each iteration, we pretrain on TCGA for 25 epochs and fine-tune a new labeler on Pannuke for 100 epochs. This process is repeated four times for a total of 100 pretraining epochs. When freezing the encoder, we see no difference between cyclic and standard pretraining after 100 epochs. When fully fine-tuning, however, we see that cyclic training surpasses standard training in only 50 epochs and after four iterations exceeds ImageNet supervised

by 0.3% (+0.0013 PQ).

We believe that the main reason for cyclic training's success comes from its ability to escape local minima with its curriculum learning framework. While we could attribute these improvements to simply using more accurate pseudo-labels, we will explore why we do not believe this is actually the case in Section 4.4.2.

4.3 Point Similarity Visualizations

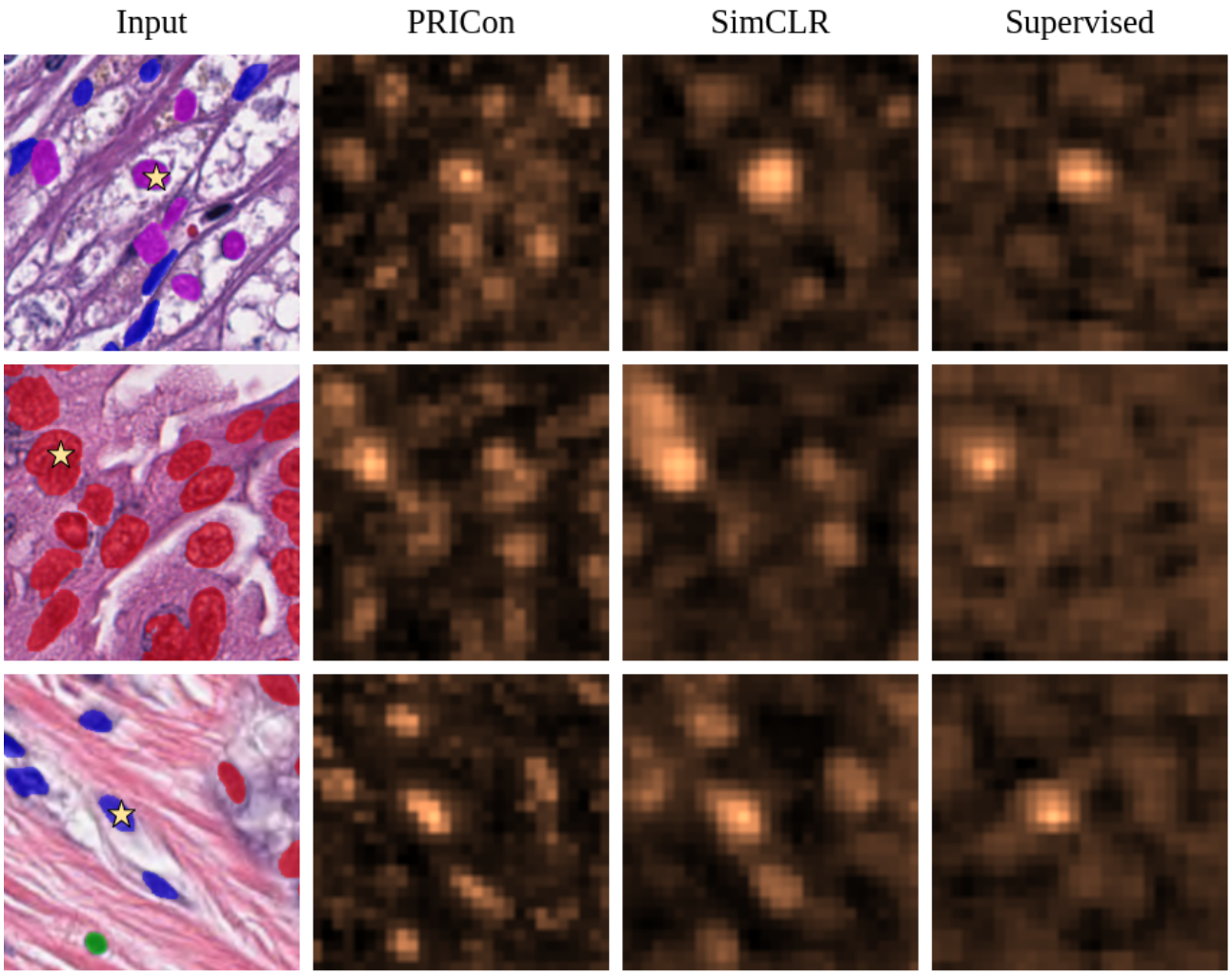


Figure 4.2: Example point similarity visualizations from pretrained encoder. From left to right, the input image overlaid with the ground truth segmentation mask (background class is not shown for better visibility), similarity matrices from PRICon, SimCLR and ImageNet supervised encoders. We mark anchor points with a star. Brighter pixels are more similar to the anchor point.

To better understand what the PRICon objective learns, we visualize the point similarities of the pretrained encoder following the approach used by Bai et al. [6]. We pass 896×896 resolution images through the pretrained encoder to get dense feature maps with a spatial resolution of 28×28. This high resolution is used to obtain more informative and precise visualizations compared to the standard 7×7 feature maps. We then measure the cosine similarity of a chosen anchor point with all other points in the feature map to produce a similarity matrix. For improved visibility, we upsample the similarity matrix to 224×224

resolution. We use images from the Pannuke dataset in order to have corresponding ground truth segmentation masks that can be used for comparison.

We visualize example similarity matrices from PRICon, SimCLR, and ImageNet supervised pretrained encoders in Figure 4.3. We see that PRICon's similarity matrices line up closer to the ground truth segmentation masks compared to the SimCLR and supervised models. SimCLR tends to gradually diminish the similarity around the center of a nucleus, causing it to overestimate the background region surrounding it. Because of PRICon's point and region-level objectives, it produces much more precise similarity matrices, placing high similarity on top of nuclei and quickly switching to low similarity once it reaches the background, making it better align the ground truth masks. PRICon also has better coverage of all nuclei, whereas SimCLR misses many, especially those far away from the anchor point.

In Figure 4.3, we select multiple anchor points from different ground truth class types to see how the similarity matrices change. When we select anchor points from different nuclei classes, we see that PRICon gives higher similarity to nuclei of the same class type compared to the other classes. SimCLR also exhibits this property to some extent but tends to concentrate on the nucleus that the anchor point belongs to. When we select a point from the background region, SimCLR only places high similarity around the anchor point and fails to identify the rest of the background region in the image. PRICon, on the other hand, identifies the background region across the entire image and accurately places low similarity at the positions of each nucleus. We believe this strong performance on background regions comes from the fact that the background class is present in every training image which causes it to be sampled more often in the point and region-level losses and teaches the encoder to better discriminate background pixels. This suggests that incorporating a better sampling strategy for the less common classes may improve discrimination between nuclei types which we leave for future work.

4.4 Ablation Study

Unless specified, in all ablation study experiments we pretrain ResNet-18 encoders with PRICon on the TCGA dataset for 25 epochs using a labeler trained on the Pannuke dataset and fine-tune on Pannuke for 100 epochs.

$\mathcal{L}_{pt}$	$\mathcal{L}_{reg}$	$\mathcal{L}_{img}$	**PQ**
✓			0.3665
	✓		0.3547
		✓	0.3466
✓		✓	0.3625
	✓	✓	0.3621
✓	✓		0.3636
✓	✓	✓	**0.3769**

(a) Loss functions.

N	**PQ**
8	0.3714
16	0.3694
32	**0.3769**
64	0.3727

(b) Point-level loss number of sampled points.

N	**R**	**PQ**
16	7×7	0.3694
16	14×14	0.3753
16	28×28	0.3733
16	56×56	0.3720
32	7×7	**0.3769**
32	14×14	0.3764
32	28×28	0.3677

(c) Dense representations resolution.

Table 4.6: Results of the hyperparameter ablation study. We pretrain for 25 epochs and report the average panoptic quality across three data folds in the frozen encoder fine-tuning setting. We ablate on the (a) PRICon loss function components, (b) number of sampled points in the point-level loss and (c) the spatial resolution of the dense image representations. Blue denotes PRICon's default settings.

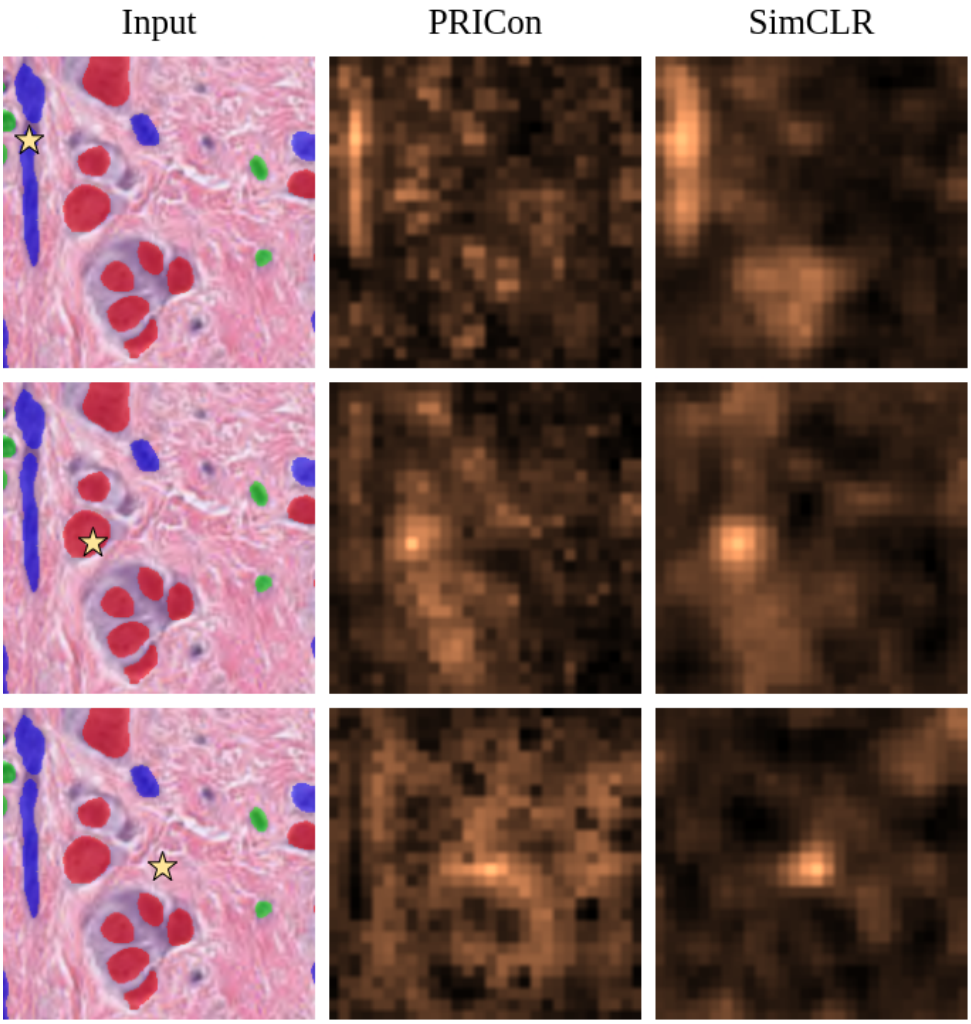


Figure 4.3: Example point similarity visualizations from multiple anchor points. From left to right, the input image overlaid with the ground truth segmentation mask (background class is not shown for better visibility), similarity matrices from PRICon and SimCLR encoders. In the first and second rows, we select anchor points from nuclei belonging to different classes and in the third row, we select an anchor point from the background. We denote anchor points with a star. Brighter pixels are more similar to the anchor point.

4.4.1 Hyperparameters

In the hyperparameter ablation study experiments, we report frozen encoder fine-tuning results as it provides better insights on how the changes impact models' ability to learn the PRICon objective.

Loss Functions

To understand the importance of each loss function, we pretrain with PRICon using different combinations of the point ($\mathcal{L}_{pt}$), region ($\mathcal{L}_{reg}$) and image-level ($\mathcal{L}_{img}$) losses. We report the results in Table 4.6a. We see that using all three losses achieves the best results, verifying the importance of each objective. Optimizing only the point-level objective achieves the next best results and incorporating only one of the region or image-level reduces performance. We would like to note that tuning the loss weights for each combination of losses would likely improve results.

Number of Points

For the point-level loss, we experiment with the number of sampled points N and report the results in Table 4.6b. While not monotonically increasing, we do see the performance improve as we sample up to $N = 32$ points before decreasing. Similar to how SimCLR improves with a large batch size [15, 16], a large N likely also improves performance by increasing the number of samples used during the point-level loss calculation. Unlike increasing the batch size, however, increasing N does not significantly increase memory requirements as we do not need to pass additional batch elements through the encoder each iteration. We can likely gain further improvements from scaling N by lowering the learning rate and lengthening the training schedule [15, 16, 33].

Feature Resolution

Prior dense contrastive approaches have observed performance improvements by increasing the resolution of the dense feature representations using interpolation [76, 6]. We test this idea by upsampling the 7×7 representations using bilinear interpolation before calculating the PRICon loss and report the results in Table 4.6c. Increasing the representation's resolution gives mixed results. When we sample the default number of points, $N = 32$, upsampling does not improve performance and using the original 7×7 features achieves the best results. However, when we set $N = 16$, increasing to 14×14 resolution improves performance by a sizable margin and then worsens as we upsample further.

4.4.2 Labeler Accuracy

To assess how the quality of the pseudo-labels impacts PRICon's performance, we pretrain encoders with labelers of varying accuracy and measure their fine-tuning performance. We plot the results in Figure 4.4 and report exact values in Table A.4. When fine-tuning only the decoder, we see that a more accurate labeler leads to improved fine-tuning performance, increasing by as much as 7.0% (+0.0248 PQ). When fine-tuning the entire segmentation model, however, we see that the accuracy of the labeler has little impact on fine-tuning results. Apart from when the labeler's PQ is < 0.1, full fine-tuning consistently stays between $0.392 - 0.394$ PQ and shows no sign of any further improvements with a stronger labeler. This robustness to pseudo-label quality demonstrates that PRICon can be applied in use-cases where training a strong labeler is not possible (e.g. limited labeled data) and suggests that using an unsupervised segmentation algorithm may also be a possible alternative to make PRICon fully self-supervised.

4.4.3 Choice of Labeler Dataset

By default, we use a labeler trained on the same dataset that we later fine-tune on. To understand the importance of this choice, we pretrain two encoders using labelers trained on Pannuke and Lizard, respectively, and then fine-tune each encoder on Pannuke, Pannuke-Breast and Lizard. For these experiments, we pretrain for 100 epochs. We report the results in Table 4.7. As we would expect, when freezing the encoder, a mismatch between the labeler and fine-tuning datasets results in a noticeable performance drop, since the encoder cannot unlearn any dataset biases inherited from the labeler during pretraining. When fine-tuning the entire model on Pannuke and Lizard, we see that a dataset mismatch does not impact performance but actually gives a slight improvement in both cases. However, on Pannuke-Breast we see a performance drop when using Lizard for the labeler dataset which may be caused by the lack of cancer type overlap between Lizard and Pannuke-Breast which have colon and breast cancer images, respectively.

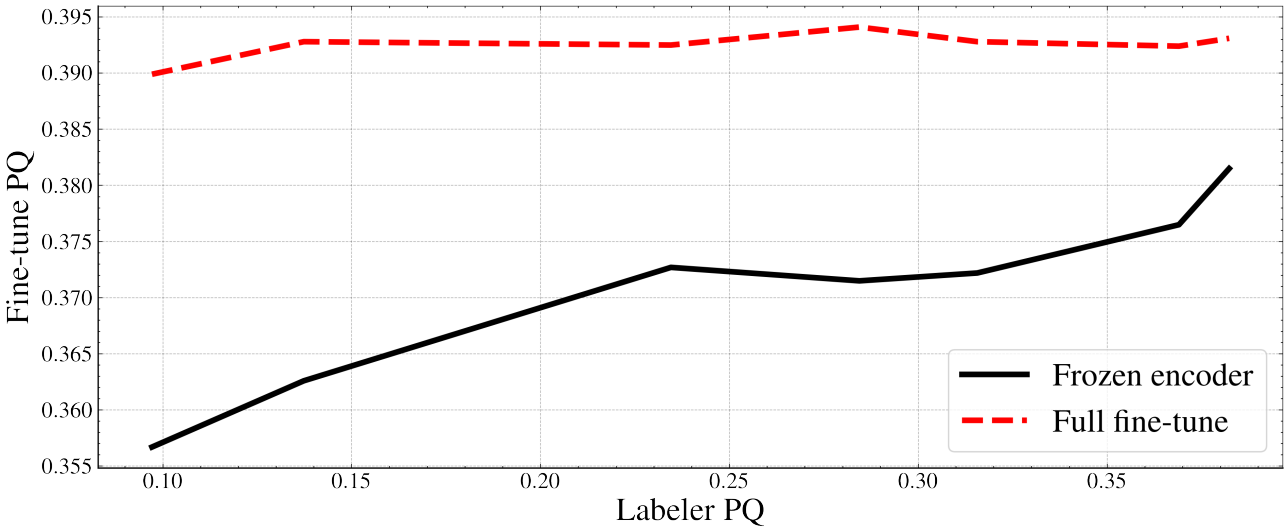


Figure 4.4: Results of the labeler accuracy ablation study. We pretrain with PRICon using labelers of varying accuracy and report fine-tuning results on the Pannuke dataset. The x-axis denotes the labeler's panoptic quality on Pannuke and the y-axis denotes the fine-tuning panoptic quality of the pretrained model.

Fine-tuning Dataset	**Labeler Dataset**	**PQ**	
		Frozen Encoder	Full Fine-tuning
Pannuke	Pannuke	**0.3815**	0.3931
Pannuke	Lizard	0.3735	**0.3941**
Pannuke-Breast	Pannuke	**0.4019**	**0.4300**
Pannuke-Breast	Lizard	0.3908	0.4215
Lizard	Lizard	**0.2785**	0.3474
Lizard	Pannuke	0.2597	**0.3486**

Table 4.7: Results of the labeler and fine-tuning dataset ablation study. *Fine-tuning dataset* is the dataset that the pretrained encoder is fine-tuned on. *Labeler dataset* is the dataset that PRICon's labeler is trained on.

4.4.4 ImageNet Weights Initialization

Prior works in pretraining for medical imaging have observed improved performance when starting from ImageNet pretrained weights and continuing to pretrain with an in-domain medical image dataset [3, 2]. We similarly experiment with this by initializing the encoder with ImageNet supervised weights and continue pretraining with PRICon on the TCGA dataset. We report the results in Table 4.8. When we pretrain with PRICon for 25 epochs, the mixed model improves on both of supervised and PRICon's weak points, improving 4.0% (+0.0145 PQ) over supervised pretraining in frozen encoder fine-tuning and 0.7% (+0.0026 PQ) over PRICon in full fine-tuning. In both settings, however, the mixed model does not achieve the overall best results, falling between the fully supervised and PRICon models. When we pretrain for 100 epochs, full fine-tuning performance begins to deteriorate. We suspect that this is likely due

Method	TCGA Epochs	PQ	
		Frozen Encoder	Full Fine-tuning
Supervised	0	0.3641	**0.3977**
PRICon	100	**0.3815**	0.3931
Supervised → PRICon	25	0.3786	0.3957
Supervised → PRICon	100	0.3779	0.3907

Table 4.8: Results of pretraining with ImageNet supervised initialized weights. *Supervised → PRICon* denotes beginning PRICon pretraining with ImageNet supervised encoder weights. *TCGA epochs* is the number of completed pretraining epochs on the TCGA dataset.

to the model unlearning most of supervised pretraining's features and falling into a bad local minimum. Compared to the prior works which found substantial improvements with this pretraining strategy [3, 2], we believe our less impressive results come from the different biases both pretraining approach learns, with supervised pretraining being biased to classification and PRICon being biased to segmentation, causing them to not exhibit any sort of compounding effect.

Chapter 5

Conclusion

In this work, we introduced the Point Region Image-level Contrast pretraining framework. PRICon simultaneously optimizes point, region and image-level contrastive loss functions to learn rich image representations from unlabeled data which transfers well to dense prediction tasks. By using a trained labeler network to generate pseudo-label masks for unlabeled data, PRICon can generate a highly diverse and accurate set of positive and negative contrastive pairs to learn task-specific image representations. To demonstrate the advantages of task-specific pretraining, we applied PRICon to the histopathology domain by pretraining on unlabeled whole slide images and fine-tuning on various nucleus segmentation datasets. We found that PRICon is more effective at pretraining on unlabeled histopathology images than prior self-supervised and semi-supervised approaches, being competitive with ImageNet supervised pretraining, and that PRICon trained models are significantly more data efficient when fine-tuning on limited labeled data. We also proposed a cyclic bootstrapping strategy to gradually improve pseudo-labels during pretraining for improved downstream performance. Through our ablation study, we found that PRICon is robust to the quality of the pseudo-labels, making the approach feasible in scenarios where training a strong labeler is not possible.

While we have identified multiple benefits of PRICon's task-specific pretraining approach, it does have drawbacks in terms of generalization and sensitivity to experimental design choices. Future work will look into approaches to improve on these drawbacks, such as incorporating uncertainty into the pseudo-labels, using a more diverse set of pseudo-label classes and generating masks with unsupervised algorithms to make PRICon fully self-supervised. The PRICon objective is just one example of using pseudo-label guidance to learn task-specific image representations. We believe that designing loss functions for specific downstream tasks is a promising direction to help bridge the gap between the pretraining and fine-tuning stages.

Appendix A

Appendix

A.1 Data Augmentation Parameters

In Table A.1, we provide the parameter values for the data augmentations used for PRICon pretraining. Color jitter consists of the combination of brightness, contrast, saturation and hue distortions.

During segmentation fine-tuning, we use random Gaussian blur with a kernel size sampled between $[3, 7]$, Gaussian noise with a variance sampled between $[0, 25]$, random rotation in increments of 90°, and horizontal and vertical flips. All augmentations are applied with a probability of 0.5. We train on the original 256×256 image patches.

Parameter	**Value**
Random crop probability	1.0
Crop size	224
Max crop scale	1.0
Min crop scale	0.2
Horizontal flip probability	0.5
Rotation probability	0.5
Color jitter probability	0.8
Brightness intensity	0.8
Contrast intensity	0.8
Saturation intensity	0.8
Hue intensity	0.2
Grayscale conversion probability	0.2
Gaussian blur probability	0.0

Table A.1: Data augmentation parameter values used during PRICon pretraining.

A.2 Detailed Baseline Settings

Supervised We use the ImageNet supervised classification pretrained weights provided in the PyTorch library [63].

SimCLR [15, 16] We pretrain a ResNet-18 [40] encoder without the final classification layer and use an MLP projection head which consists of two linear layers with a hidden dimension of 2048 and an output dimension of 128. We apply a ReLU activation between the linear layers.

For data augmentations, we use the same augmentations and parameter values used in PRICon outlined in Section 4.1.2 and Table A.1. We optimize using LARS [84] with a batch size of 256 and weight decay of 10^{-4}. We exclude the bias and batch normalization parameters from the LARS adaptation. The initial learning rate is 0.2 and we decay it to 0 following a cosine annealing schedule [58] for 100 epochs on the TCGA dataset. For the contrastive loss, we use a temperature of 0.5. We train using mixed precision [59].

SimSiam [17] We pretrain a ResNet-18 encoder without the final classification layer. The MLP projection head consists of three linear layers with hidden and output dimensions of 2048. We apply batch normalization [45] and a ReLU activation between the linear layers and only apply batch normalization after the final layer. The MLP prediction head consists of two linear layers with a hidden dimension of 512 and an output dimension of 2048. We apply batch normalization and a ReLU activation between the linear layers. For data augmentations, we use the same augmentations outlined in Section 4.1.2. We apply weak color jitter which has a brightness, contrast and saturation intensity of 0.4 and a hue intensity of 0.1. All other augmentation parameters are the same as in Table A.1. For optimization, we use stochastic gradient descent with a batch size of 256 and a weight decay of 10^{-4}. The initial learning rate is 0.075 and we decay it to 0 following a cosine annealing schedule for 100 epochs on the TCGA dataset except for the prediction head which is kept fixed at the initial learning rate [17]. We train using mixed precision.

CutMix [25] We train a U-Net [66] segmentation model with a ResNet-18 encoder. For data augmentations, we use random Gaussian noise with a variance between $[0, 25]$, rotations in increments of 90° and horizontal flips. All augmentations are applied with a probability of 0.5. For the CutMix augmentation, the window is a rectangle that covers up to 50% of the original image. For optimization, we use Adam [48] with a β_1 of 0.9, β_2 of 0.999, batch size of 4 and a weight decay of $5 \cdot 10^{-4}$. The initial learning rate is $5 \cdot 10^{-4}$ and we decay it to 0 following a cosine annealing schedule for 80,000 iterations. The teacher network is updated by an exponential moving average with a decay of 0.99. For unlabeled data, we use the TCGA dataset and for labeled data, we use the target fine-tuning dataset.

ClassMix [62] We use the same data augmentations and optimization procedure as used in CutMix. For the ClassMix augmentation, the mixing mask covers a random selection of half the object classes present in an image.

ReCo [55] We use the same data augmentations and optimization procedure as CutMix except we use an initial learning rate of 10^{-4}. We use ClassMix for the consistency augmentation. For the ReCo loss, we use a weak threshold of 0.7, a strong threshold of 0.97, a temperature of 0.5 and sample 512 negative points and 256 query points.

A.3 Additional Results

A.3.1 TNBC

The TNBC dataset [60] is a small-scale binary nucleus segmentation dataset which contains 50 patches of size 512×512 from breast cancer slides. In our experiments, we split the dataset into 43 training images and 7 test images. For PRICon, we experiment with using labelers trained on TNBC and on Pannuke and pretrain for 100 epochs on TCGA. We use the same fine-tuning settings as the previous experiments, except the initial learning rate is $5 \cdot 10^{-5}$ and

Method	Dice	
	Frozen Encoder	Full Fine-tuning
Scratch	-	0.8324
Supervised (ImageNet)	**0.8235**	**0.8418**
SimCLR [15]	0.8165	0.8383
SimSiam [17]	0.7917	0.8413
CutMix [25]	-	0.8190
ClassMix [62]	-	0.8160
ReCo [55]	-	0.8229
PRICon (Pannuke)	0.8172	**0.8420**
PRICon (TNBC)	0.8071	0.8410

Table A.2: Results on the TNBC dataset. We fine-tune a U-Net segmentation network for 1000 iterations and report the average dice score on the test set across five independent runs. For PRICon, we pretrain encoders using a labeler trained on Pannuke and on TNBC.

we fine-tune for 1000 iterations. In Table A.2, we report the average dice score on the TNBC dataset across five independent fine-tuning runs.

When fine-tuning only the decoder, supervised pretraining performs the best while PRICon achieves similar performance to SimCLR. For full fine-tuning, PRICon is equal in performance to supervised pretraining. Training the labeler on Pannuke outperforms training on TNBC which we speculate is because TNBC is only a binary dataset, making the contrastive objectives easier and less informative.

A.3.2 ResNet-50 Encoder

Method	PQ	
	Frozen Encoder	Full Fine-tuning
Scratch	-	0.3718
Supervised	0.3847	**0.4041**
PRICon	**0.3891**	0.3989

Table A.3: Results on the Pannuke dataset using pretrained ResNet-50 encoders. We report the average cross-validation panoptic quality across three data folds.

We pretrain a ResNet-50 encoder with PRICon for 100 epochs and report the results of fine-tuning U-Net segmentation models with a ResNet-50 encoder on the Pannuke dataset in Table A.3. The results are similar to the ResNet-18 results, with PRICon outperforming supervised in the frozen encoder setting by 1.1% (+0.0044 PQ) but falling to supervised in full fine-tuning 1.3% (-0.0052 PQ). We do note that the performance gap between PRICon and supervised in the frozen encoder setting is smaller than with ResNet-18 encoders.

A.3.3 Labeler Accuracy

In Table A.4, we report the specific labeler and fine-tuning PQ values on the Pannuke dataset from the experiments in Section 4.4.2 and plotted in Figure 4.4.

Labeler PQ	**PQ**	
	Frozen Encoder	Full Fine-tuning
0.0971	0.3567	0.3899
0.1373	0.3626	0.3928
0.2345	0.3727	0.3925
0.2844	0.3715	0.3941
0.3156	0.3722	0.3928
0.3690	0.3765	0.3924
0.3823	0.3815	0.3931

Table A.4: Results on the Pannuke dataset from the labeler accuracy ablation study.

Bibliography

[1] Laith Alzubaidi, Mohammed A Fadhel, Omran Al-Shamma, Jinglan Zhang, J Santamaría, Ye Duan, and Sameer R. Oleiwi. Towards a better understanding of transfer learning for medical imaging: a case study. *Applied Sciences*, 10(13):4523, 2020.

[2] Shekoofeh Azizi, Laura Culp, Jan Freyberg, Basil Mustafa, Sebastien Baur, Simon Kornblith, Ting Chen, Patricia MacWilliams, S Sara Mahdavi, Ellery Wulczyn, et al. Robust and efficient medical imaging with self-supervision. *arXiv preprint arXiv:2205.09723*, 2022.

[3] Shekoofeh Azizi, Basil Mustafa, Fiona Ryan, Zachary Beaver, Jan Freyberg, Jonathan Deaton, Aaron Loh, Alan Karthikesalingam, Simon Kornblith, Ting Chen, et al. Big self-supervised models advance medical image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3478–3488, 2021.

[4] Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. *Advances in neural information processing systems*, 27, 2014.

[5] Wenjia Bai, Chen Chen, Giacomo Tarroni, Jinming Duan, Florian Guitton, Steffen E Petersen, Yike Guo, Paul M Matthews, and Daniel Rueckert. Self-supervised learning for cardiac mr image segmentation by anatomical position prediction. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 541–549. Springer, 2019.

[6] Yutong Bai, Xinlei Chen, Alexander Kirillov, Alan Yuille, and Alexander C Berg. Point-level region contrast for object detection pre-training. *arXiv preprint arXiv:2202.04639*, 2022.

[7] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.

[8] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.

[9] Joseph Boyd, Mykola Liashuha, Eric Deutsch, Nikos Paragios, Stergios Christodoulidis, and Maria Vakalopoulou. Self-supervised representation learning using visual field expansion on digital pathology. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 639–647, 2021.

[10] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[11] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers.

In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021.

[12] Hao Chen, Qi Dou, Xi Wang, Jing Qin, and Pheng Ann Heng. Mitosis detection in breast cancer histology images via deep cascaded networks. In *Thirtieth AAAI conference on artificial intelligence*, 2016.

[13] Kai Chen, Lanqing Hong, Hang Xu, Zhenguo Li, and Dit-Yan Yeung. Multisiam: Self-supervised multi-instance siamese representation learning for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7546–7554, 2021.

[14] Sihong Chen, Kai Ma, and Yefeng Zheng. Med3d: Transfer learning for 3d medical image analysis. *arXiv preprint arXiv:1904.00625*, 2019.

[15] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[16] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020.

[17] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

[18] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9640–9649, 2021.

[19] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

[20] Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. *Advances in neural information processing systems*, 33:8765–8775, 2020.

[21] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[22] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[23] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015.

[24] Aleksandr Ermolov, Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening for self-supervised representation learning. In *International Conference on Machine Learning*, pages 3015–3024. PMLR, 2021.

[25] Geoff French, Samuli Laine, Timo Aila, Michal Mackiewicz, and Graham Finlayson. Semi-supervised semantic segmentation needs strong, varied perturbations. *arXiv preprint arXiv:1906.01916*, 2019.

[26] Jevgenij Gamper, Navid Alemi Koohbanani, Ksenija Benet, Ali Khuram, and Nasir Rajpoot. Pannuke: an open pan-cancer histology dataset for nuclei instance segmentation and classification. In *European Congress on Digital Pathology*, pages 11–19. Springer, 2019.

[27] Jevgenij Gamper, Navid Alemi Koohbanani, Ksenija Benes, Simon Graham, Mostafa Jahanifar, Syed Ali Khurram, Ayesha Azam, Katherine Hewitt, and Nasir Rajpoot. Pannuke dataset extension, insights and baselines. *arXiv preprint arXiv:2003.10778*, 2020.

[28] Songwei Ge, Shlok Mishra, Chun-Liang Li, Haohan Wang, and David Jacobs. Robust contrastive learning using negative samples with diminished semantics. *Advances in Neural Information Processing Systems*, 34, 2021.

[29] Priya Goyal, Quentin Duval, Isaac Seessel, Mathilde Caron, Mannat Singh, Ishan Misra, Levent Sagun, Armand Joulin, and Piotr Bojanowski. Vision models are more robust and fair when pretrained on uncurated images without supervision. *arXiv preprint arXiv:2202.08360*, 2022.

[30] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and benchmarking self-supervised visual representation learning. In *Proceedings of the ieee/cvf International Conference on computer vision*, pages 6391–6400, 2019.

[31] Simon Graham, Mostafa Jahanifar, Ayesha Azam, Mohammed Nimir, Yee-Wah Tsang, Katherine Dodd, Emily Hero, Harvir Sahota, Atisha Tank, Ksenija Benes, et al. Lizard: A large-scale dataset for colonic nuclear instance segmentation and classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 684–693, 2021.

[32] Simon Graham, Quoc Dang Vu, Shan E Ahmed Raza, Ayesha Azam, Yee Wah Tsang, Jin Tae Kwak, and Nasir Rajpoot. Hover-net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images. *Medical Image Analysis*, 58:101563, 2019.

[33] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020.

[34] Robert L Grossman, Allison P Heath, Vincent Ferretti, Harold E Varmus, Douglas R Lowy, Warren A Kibbe, and Louis M Staudt. Toward a shared vision for cancer genomic data. *New England Journal of Medicine*, 375(12):1109–1112, 2016.

[35] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010.

[36] Michael U Gutmann and Aapo Hyvärinen. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of machine learning research*, 13(2), 2012.

[37] Guy Hacohen and Daphna Weinshall. On the power of curriculum learning in training deep networks. In *International Conference on Machine Learning*, pages 2535–2544. PMLR, 2019.

[38] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

[39] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[40] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[41] Olivier J Hénaff, Skanda Koppula, Jean-Baptiste Alayrac, Aaron van den Oord, Oriol Vinyals, and João Carreira. Efficient visual pretraining with contrastive detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10086–10096, 2021.

[42] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

[43] Tianyu Hua, Wenxiao Wang, Zihui Xue, Sucheng Ren, Yue Wang, and Hang Zhao. On feature decorrelation in self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9598–9608, 2021.

[44] Tri Huynh, Simon Kornblith, Matthew R Walter, Michael Maire, and Maryam Khademi. Boosting contrastive self-supervised learning with false negative cancellation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2785–2795, 2022.

[45] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[46] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):4037–4058, 2020.

[47] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[48] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[49] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9404–9413, 2019.

[50] Navid Alemi Koohbanani, Balagopal Unnikrishnan, Syed Ali Khurram, Pavitra Krishnaswamy, and Nasir Rajpoot. Self-path: Self-supervision for classification of pathology images with limited annotations. *IEEE Transactions on Medical Imaging*, 40(10):2845–2856, 2021.

[51] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2661–2671, 2019.

[52] Jiajun Li, Tiancheng Lin, and Yi Xu. Sslp: Spatial guided self-supervised learning on pathological images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 3–12. Springer, 2021.

[53] Xiaoni Li, Yu Zhou, Yifei Zhang, Aoting Zhang, Wei Wang, Ning Jiang, Haiying Wu, and Weiping Wang. Dense semantic contrast for self-supervised visual representation learning. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 1368–1376, 2021.

[54] Gaobo Liang and Lixin Zheng. A transfer learning method with deep residual network for pediatric pneumonia diagnosis. *Computer methods and programs in biomedicine*, 187:104964, 2020.

[55] Shikun Liu, Shuaifeng Zhi, Edward Johns, and Andrew J Davison. Bootstrapping semantic segmentation with regional contrast. *arXiv preprint arXiv:2104.04465*, 2021.

[56] Songtao Liu, Zeming Li, and Jian Sun. Self-emd: Self-supervised object detection without imagenet. *arXiv preprint arXiv:2011.13677*, 2020.

[57] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12009–12019, 2022.

[58] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[59] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. Mixed precision training. *arXiv preprint arXiv:1710.03740*, 2017.

[60] Peter Naylor, Marick Laé, Fabien Reyal, and Thomas Walter. Nuclei segmentation in histopathology images using deep neural networks. In *2017 IEEE 14th international symposium on biomedical imaging (ISBI 2017)*, pages 933–936. IEEE, 2017.

[61] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016.

[62] Viktor Olsson, Wilhelm Tranheden, Juliano Pinto, and Lennart Svensson. Classmix: Segmentation-based data augmentation for semi-supervised learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1369–1378, 2021.

[63] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[64] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

[65] Christian S Perone and Julien Cohen-Adad. Deep semi-supervised segmentation with weight-averaged consistency targets. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pages 12–19. Springer, 2018.

[66] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[67] Yoni Schirris, Efstratios Gavves, Iris Nederlof, Hugo Mark Horlings, and Jonas Teuwen. Deepsmile: Self-supervised heterogeneity-aware multiple instance learning for dna damage response defect classification directly from h&e whole-slide images. *arXiv preprint arXiv:2107.09405*, 2021.

[68] Karin Stacke, Jonas Unger, Claes Lundström, and Gabriel Eilertsen. Learning representations with contrastive self-supervised learning for histopathology applications. *arXiv preprint arXiv:2112.05760*, 2021.

[69] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017.

[70] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

[71] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *Advances in Neural Information Processing Systems*, 33:6827–6839, 2020.

[72] Yuandong Tian, Xinlei Chen, and Surya Ganguli. Understanding self-supervised learning dynamics without contrastive pairs. *CoRR*, abs/2102.06810, 2021.

[73] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

[74] Feng Wang, Tao Kong, Rufeng Zhang, Huaping Liu, and Hang Li. Self-supervised learning by estimating twin class distributions. *arXiv preprint arXiv:2110.07402*, 2021.

[75] Xiang Wang, Xinlei Chen, Simon S. Du, and Yuandong Tian. Towards demystifying representation learning with non-contrastive self-supervision, 2021.

[76] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3024–3033, 2021.

[77] Fangyun Wei, Yue Gao, Zhirong Wu, Han Hu, and Stephen Lin. Aligning pretraining for detection via object-level contrastive learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[78] Tete Xiao, Colorado J Reed, Xiaolong Wang, Kurt Keutzer, and Trevor Darrell. Region similarity representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10539–10548, 2021.

[79] Huidong Xie, Hongming Shan, Wenxiang Cong, Xiaohua Zhang, Shaohua Liu, Ruola Ning, and Ge Wang. Dual network architecture for few-view ct-trained on imagenet data and transferred for medical imaging. In *Developments in X-ray Tomography XII*, volume 11113, pages 184–194. SPIE, 2019.

[80] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698, 2020.

[81] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16684–16693, 2021.

[82] Pengshuai Yang, Zhiwei Hong, Xiaoxu Yin, Chengzhan Zhu, and Rui Jiang. Self-supervised visual representation learning for histopathological images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 47–57. Springer, 2021.

[83] Chun-Hsiao Yeh, Cheng-Yao Hong, Yen-Chi Hsu, Tyng-Luh Liu, Yubei Chen, and Yann LeCun. Decoupled contrastive learning. *arXiv preprint arXiv:2110.06848*, 2021.

[84] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017.

[85] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021.

[86] X Zhai, A Kolesnikov, N Houlsby, and L Beyer. Scaling vision transformers. arxiv 2021. *arXiv preprint arXiv:2106.04560*.

[87] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016.

[88] Xiaojin Jerry Zhu. Semi-supervised learning literature survey. 2005.

[89] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc Le. Rethinking pre-training and self-training. *Advances in neural information processing systems*, 33:3833–3845, 2020.