

An exploratory study of self-supervised pre-training on partially supervised multi-label classification on chest X-ray images

Nanqing Dong^{a,*}, Michael Kampffmeyer^{b,*}, Haoyang Su^a, Eric Xing^{c,d}

^a Shanghai Artificial Intelligence Laboratory, Shanghai, 200232, China

^b Department of Physics and Technology, UiT The Arctic University of Norway, Tromsø, 9019, Norway

^c Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA 15213, USA

^d Mohamed bin Zayed University of Artificial Intelligence, Masdar City, Abu Dhabi, United Arab Emirates

ARTICLE INFO

Keywords:

Self-supervised learning
Partially supervised learning
Data scarcity
Multi-label classification

ABSTRACT

This paper serves as the first empirical study on self-supervised pre-training on partially supervised learning, an emerging yet unexplored learning paradigm with missing annotations. This is particularly important in the medical imaging domain, where label scarcity is the main challenge of practical applications. To promote the awareness of partially supervised learning, we leverage partially supervised multi-label classification on chest X-ray images as an instance task to illustrate the challenges of the problem of interest. Through a series of simulated experiments, the empirical findings validate that solving multiple pretext tasks jointly in the pre-training stage can significantly improve the downstream task performance under the partially supervised setup. Further, we propose a new pretext task, reverse vicinal risk minimization, and demonstrate that it provides a more robust and efficient alternative to existing pretext tasks for the instance task of interest.

1. Introduction

Fueled by the recent success of deep learning, there is a renaissance of research on self-supervised learning (SSL) [1–7]. The core concept of contemporary SSL is to formulate a *pretext* task, which has its own *annotation-free* label, *i.e.* the supervision signal of the pretext task can be defined by the information within the unlabeled data [8–12]. The goal of solving the pretext task is to learn meaningful representations for the target task of interest, *i.e.* the downstream task. It has been reported that SSL can achieve competitive performance or even outperform standard supervised learning in representation learning [8–11], and is a core component of many current medical imaging approaches [13,14]. An important application of SSL is to learn transferable representations and then fine-tune these with the labels of a given downstream task in a supervised fashion.

Though self-supervised pre-training has shown promising results in boosting the model performance under a supervised setup, especially when only limited labels are available, its role in partially supervised learning (PSL) [15] remains unclear. An illustrative diagram of self-supervised pre-training for partially supervised downstream tasks is presented in Fig. 1. PSL is an emerging label-efficient learning paradigm that has a close tie with multi-task learning (MTL) [16]. In PSL, a task of interest can be decomposed into multiple sub-tasks and each instance in the training set is labeled for a *true* subset of sub-tasks,

instead of all sub-tasks. In other words, each instance has *task-wise* missing annotations. Common downstream tasks for vision tasks that can be formulated as MTL include multi-label classification, semantic segmentation, and object detection. These tasks could all be formulated as PSL problems when multiple relatively small datasets annotated for specific downstream tasks are collected to form a larger dataset.

The partial label problem is a common challenge in medical image analysis. As an exploratory study in this field, this work utilizes partially supervised multi-label classification (PSMLC) [17] on chest X-ray images (CXRs) as the downstream task to demonstrate the impact of self-supervised pre-training on PSL. PSMLC is a partially supervised variant of standard multi-label classification (MLC), where the training data do not have complete annotations for all classes of interest. For example, a small pneumonia dataset and a small tuberculosis dataset are collected from two different hospitals and annotated by clinicians with different expertise to form a large dataset. Each medical image is then partially labeled concerning either pneumonia or tuberculosis. An example is presented in Fig. 2 to convey the concept of PSMLC. Specifically, we use thoracic disease classification on chest X-ray images (CXRs) to illustrate two practical challenges in PSL [18], namely label scarcity and inter-class imbalance. Given limited partially labeled CXR

* Corresponding authors.

E-mail addresses: dongnanqing@pjlab.org.cn (N. Dong), michael.c.kampffmeyer@uit.no (M. Kampffmeyer).

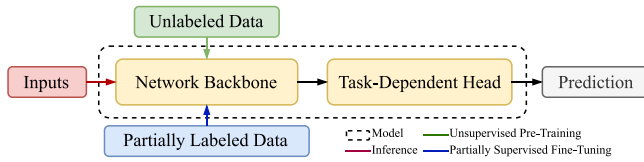
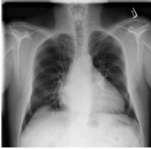


Fig. 1. Diagram of self-supervised pre-training for partially supervised downstream tasks. The network backbone (feature extractor) is pre-trained on unlabeled data. Then, the network backbone and task-dependent head are jointly fine-tuned with the partially supervised data.

Fully Labeled 	Atelectasis	0
	Cardiomegaly	1
	Consolidation	0
	Emphysema	1
	Effusion	0
	Fibrosis	0
	Hernia	0


Partially Labeled 	Atelectasis	0
	Cardiomegaly	?
	Consolidation	?
	Emphysema	?
	Effusion	?
	Fibrosis	?
	Hernia	?

Fig. 2. Partially supervised multi-label classification on medical images. The top row illustrates a fully labeled CXR with respect to diseases of interest, while the bottom row illustrates a partially labeled one. “1” and “0” denote whether a disease is diagnosed, and “?” denotes the missing annotation.

Source: The images are taken from the ChestX-ray14 dataset [19].

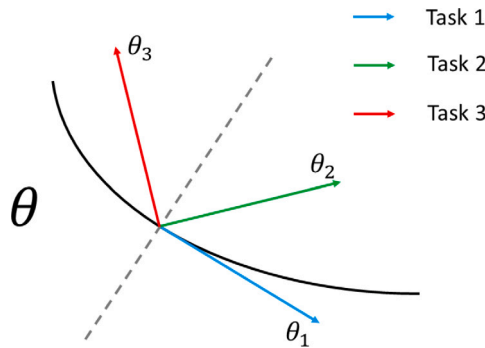


Fig. 3. Illustration of the concept of task affinity in terms of optimization. Intuitively, when three tasks denote three unit vectors in linear algebra, we can use the dot product between any two tasks to denote a hypothetical task affinity score between them. Let Task 1 (the blue arrowed line tangent to the arc) be the anchor task (i.e. the downstream task). The ideal pretext task should optimize the parameters θ along the direction of Task 1. Let the gray dashed line (perpendicular to Task 1) denote the zero task affinity score. Task 2 in this case should have a positive task affinity score with Task 1 as the gradient is still in a direction that contributes to a decrease of the loss. Task 3 on the other hand should have a negative task affinity score with Task 1 as the gradient direction increases the loss. However, in practice, there is no such quantitative measurement.

images, we aim to use large-scale unlabeled CXR images to learn transferable representations, which can be used to improve the performance of the downstream task.

To the best of our knowledge, there is no empirical study yet on the problem of interest. In this study, we try to answer three exploratory research questions in order to facilitate the understanding of the impact of self-supervised pre-training on partially supervised downstream tasks. **RQ1:** Is *task affinity* a reliable tool to choose an efficient pretext task

for PSMLC on CXRs? **RQ2:** What is the impact of self-supervised pre-training on PSMLC on CXRs? **RQ3:** Can MTL improve self-supervised pre-training when PSMLC on CXRs is the downstream task?

In this work, the discussion of task affinity is bounded within the scope of machine learning (ML). The concept arises from the study of MTL [16], which aims to understand which tasks should be learned together to improve overall learning performance [20,21]. Then, it is conceptualized as a term to describe the task alignment between the source and target tasks in transfer learning [22], i.e. the pretext and downstream tasks in the literature of SSL. As the task affinity between the pretext and downstream tasks cannot be directly measured, we use *proxy task affinity* for quantitative comparison. Here, we use the term “proxy” following the definition in SSL literature [8,9,11], where the performance of the pretext task cannot be assessed quantitatively. The “proxy task” instead is a task that is defined in such a fashion: the learning outcomes of the pretext task (e.g. representations) can be linked to a task that can directly be quantitatively measured.

In MTL, grouping tasks with large task affinities can efficiently improve the training performance [21]. Thus, a larger task affinity between the pretext task and the downstream task should intuitively lead to better transfer learning performance. A few representative pretext tasks are discussed and evaluated empirically to provide insight into **RQ1**. Later in this work, we shall see that, (proxy) task affinity is not a robust measure for partially supervised downstream tasks, despite its success for fully supervised downstream tasks.

For **RQ2**, self-supervised pre-training can be viewed as providing a strong initialization for the fine-tuning phase (c.f. random initialization). Specifically, we focus on understanding the impact of self-supervised pre-training on the downstream task under partial supervision, from a task affinity perspective. A simulated experiment is designed to investigate this question in a controllable environment, where we examine three representative pretext tasks. Several interesting findings are reported. For example, we find that self-supervised pre-training does not necessarily lead to performance gain on downstream tasks with partial labels.

A recent study has shown that MTL can benefit supervised downstream tasks in self-supervised pre-training [23]. However, the role of MTL in self-supervised pre-training for partially supervised downstream tasks remains unclear. Motivated by this empirical finding, a reasonable hypothesis to **RQ3** is that multiple pretext tasks can also make a difference in self-supervised pre-training for PSL. To validate this hypothesis, multiple pretext tasks are designed based on proxy task affinity and evaluated to provide an empirical understanding of multi-pretext-task learning for PSL.

In addition, this work also presents an empirical study to explore a pretext task based on vicinal risk minimization (VRM) [24]. VRM has shown its robustness in standard supervised learning as a data augmentation technique [25]. Again, whether VRM can benefit the partially supervised downstream tasks remains unclear. Given a pair of unlabeled images and a MixUp hyperparameter [25], a vicinal image can be generated. The model of interest is pre-trained to predict the MixUp hyperparameter with two original images and the vicinal image as the input. In contrast to seminal pretext tasks, the pretext task of interest does not utilize the intrinsic information but instead utilizes an additional hyperparameter to set up the task and introduces randomness to improve generalization. The experimental results show that the proposed pretext task, *reverse vicinal risk minimization* can achieve competitive performance with seminal pretext tasks but with a smaller computational cost.

The main contributions of this work can be summarized as follows.

1. This is the first empirical study of the relationship between the pretext task and downstream task in PSL on CXRs and the first empirical study of the impact of self-supervised pre-training on PSMLC on CXRs.

2. We explore the potential of proxy task affinity in choosing an efficient pre-training pretext task for a partially supervised downstream task and empirically show the limitations of proxy task affinity.
3. This is the first study of MTL in self-supervised pre-training for PSMLC on CXRs.
4. A pretext task based on vicinal risk minimization is proposed for PSMLC on CXRs for the first time.

The rest of the paper is organized as follows. Section 2 reviews the related work on partially supervised learning and self-supervised learning. Section 3 formulates the problem of interest and Section 4 provides the preliminary knowledge to understand the pretext task. Section 5 introduces task affinity, multi-pretext learning, and reverse vicinal risk minimization. Section 6 and Section 7 provide the experimental setup and results.

2. Related work

Partially Supervised Learning PSL has become an emerging research question due to the non-trivial annotation cost in medical image analysis tasks. Instead of collecting large-scale fully labeled data, it is more convenient to collect multiple small-scale datasets that are annotated for specific sub-tasks. PSL is still in its early stage and existing studies still rely on fully labeled data [26,27] or complex specification [28,29]. Dong et al. [15] first tackles the small-scale partially labeled data with VRM. For PSMLC, Durand et al. [30] first discusses the problem and leverages large-scale partially labeled datasets to address the problem. However, this study assumes data scarcity in the fine-tuning stage, which is more realistic in the medical domain.

Self-Supervised Learning The recent renaissance of SSL is associated with pretext tasks. In the context of deep learning, by solving a different task that is related to the downstream task, the model of interest learns to extract transferable representations from the unlabeled data. These pretext tasks only utilize the information contained in the unlabeled data. However, designing a good pretext task commonly requires non-trivial human effort because a *good self-supervised task is neither simple nor ambiguous* [31]. For example, common pretext tasks include predicting the relative position information between two augmented views of the same instance [32], solving jigsaw puzzles [31], rotation prediction [33], and masked image modeling [34]. In contrast to these hand-crafted pretext tasks, the state-of-the-art SSL methods are based on instance discrimination, a pretext task also known as contrastive learning [35]. So far, contrastive learning has provided a universal pre-training solution for various downstream tasks. However, our experiments show that contrastive learning might not be an optimal solution for PSL under data scarcity.

A similar and related work to the proposed pretext task is [36], which also utilizes vicinal risk minimization [25] in SSL. There are two main differences between the proposed pretext task and [36]. First, the downstream tasks are different, where [36] targets the time series prediction task and the proposed pretext task targets PSMLC. Second, the loss functions are different with [36] leveraging a contrastive loss, which requires the processing of the examples individually. The proposed pretext task instead simply adopts the standard mean squared error (\mathcal{L}_2 loss). Further, while [36] relies on a contrastive loss to learning the representations, in the proposed pretext tasks, the model takes the concatenation of two input examples and the vicinal example and predicts the MixUp parameter [25]. See Section 5.3 for the details of the proposed pretext task.

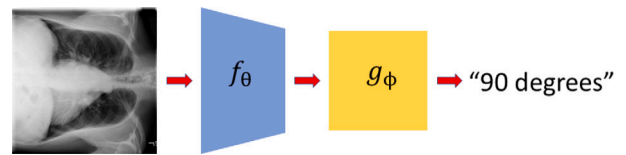


Fig. 4. Pretext task of rotation prediction (clockwise) [33]. The network is trained to predict the rotation degree.

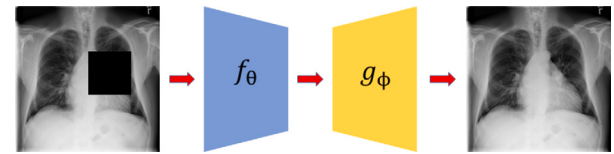


Fig. 5. Pretext task of masked image modeling [34]. The network is trained to reconstruct the masked image.

3. Problem formulation

Given a downstream task of interest, there are two training datasets, a large-scale unlabeled dataset \mathcal{U} and a small-scale partially labeled dataset \mathcal{S} . Let f_θ denote the feature extractor with parameter θ that is pre-trained on \mathcal{U} and g_ϕ^t denote the auxiliary prediction head for the pretext task t_p with parameter ϕ . In the fine-tuning phase, only f_θ will be transferred, i.e. to be fine-tuned in conjunction with a new prediction head for the downstream task.

Assume the downstream task of interest can be decomposed into a set of sub-tasks \mathcal{T} . A partially labeled training dataset \mathcal{S} consists of $K > 1$ partially labeled sub-datasets collected from K different sources, i.e. \mathcal{S} can be split into K multiple non-overlapping non-empty subsets: $\mathcal{S} = \bigcup_{i=1}^K \mathcal{D}_i$. Each subset \mathcal{D}_i is annotated with a set of tasks $\mathcal{T}_i \subset \mathcal{T}$. $\bigcup_{i=1}^K \mathcal{T}_i = \mathcal{T}$ is ensured, i.e. each sub-task has at least one instance annotated. f_θ will be fine-tuned on \mathcal{S} .

4. Preliminaries

This section provides the necessary methodological background for this study. Three representative pretext tasks for semantic understanding tasks are presented.

Let f_θ denote the feature extractor, or the model of interest, with parameter θ , and let g_ϕ^t denote the auxiliary prediction head for the pretext task t with parameter ϕ . In the fine-tuning phase, only f_θ will be transferred, i.e. to be fine-tuned in conjunction with a new prediction head for the downstream task.

4.1. Pretext tasks

4.1.1. Rotation prediction

Let x be an image and let $\tau(\cdot)$ denote a random rotation augmentation applied on x , where the rotation degree can only be chosen from $\{0, 90, 180, 270\}$ with equal probability. The pretext task is then a multi-class classification task.

$$g_\phi^{rot} \circ f_\theta : \tau(x) \mapsto \mathbb{R}^4$$

See Fig. 4 for a visual illustration of the task.

4.1.2. Masked image modeling

Let x be an image and let $\tau(\cdot)$ denote a random masking operation applied on x . The optimization goal for the pretext task is to minimize a reconstruction loss, e.g. a mean squared loss.

$$\mathcal{L}_{mim} = \|g_\phi^{mim} \circ f_\theta(\tau(x)) - x\|^2$$

See Fig. 5 for a visual illustration of the task.

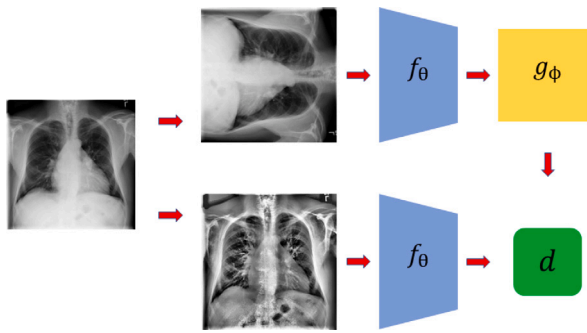


Fig. 6. Pretext task of instance discrimination. The network is trained to learn the invariance between two augmented views of the same image.

4.1.3. Instance discrimination

SimSiam [11] is chosen to illustrate the concept of instance discrimination for its simplicity and robustness.

Let $x_1 = \tau(x)$ and $x_2 = \tau'(x)$ be two augmented views of the image x . The pretext task is to maximize the agreement between the embeddings of x_1 and x_2 . Let $d(\cdot, \cdot)$ be a distance measure, the optimization goal is to minimize

$$\mathcal{L}_{ins} = d(g_\phi^{pred} \circ f_\theta(x_1), f_\theta(x_2)),$$

where g_ϕ^{pred} is a prediction head. The asymmetric design of g_ϕ^{pred} is designed to avoid collapsing solutions [10]. See Fig. 6 for a visual illustration of the task.

A common choice of $d(\cdot, \cdot)$ is the negative cosine similarity:

$$d(g_\phi^{pred} \circ f_\theta(x_1), f_\theta(x_2)) = -\frac{g_\phi^{pred} \circ f_\theta(x_1)}{\|g_\phi^{pred} \circ f_\theta(x_1)\|} \cdot \frac{f_\theta(x_2)}{\|f_\theta(x_2)\|},$$

where $\|\cdot\|$ is the norm operator.

4.2. Weighted loss

To combat class imbalance, the weighted binary cross-entropy (BCE) loss is commonly adopted in multi-label classification tasks [37]. Given an instance-label pair (x, y) , y^i denotes the i th entry of the label vector y . The weighted binary cross-entropy loss for the i th class is

$$\mathcal{L}_{BCE}(x, y^i, w_+^i, w_-^i) = -w_+^i y^i \log p(y^i = 1|x) - w_-^i (1 - y^i) \log p(y^i = 0|x), \quad (1)$$

where $w_+^i = \frac{n_-^i}{n_+^i + n_-^i}$ and $w_-^i = \frac{n_+^i}{n_+^i + n_-^i}$ with n_+^i and n_-^i being the number of positive and negative cases for the i th class, respectively.

5. Method

5.1. Task affinity

In ML, proxy task affinity has been widely adopted in MTL and SSL as quantitative tools. One of the goals of this work is to understand whether proxy task affinity is a reliable tool to study the relationship between the pretext task and the partially supervised downstream task. To provide the background, this section presents the following definitions and hypotheses.

5.1.1. Concept

So far, there is no universally good mathematical tool to quantify the similarity between tasks. In transfer learning, a high task affinity between the pretext task and the supervised downstream task is commonly a positive signal of performance gain. Thus, from the perspective of optimization, if a pretext task t_p has a high task affinity with the downstream task t_d , it should facilitate the optimization of

t_d . Concretely, at a specific time stamp in gradient descent, a random batch \mathcal{B} , and a loss measure $\mathcal{L}_{\mathcal{B}}$, the task affinity should be measured by

$$d_{ta} = \text{sim}(\nabla_{\theta_p} \mathcal{L}_{\mathcal{B}}, \nabla_{\theta_t} \mathcal{L}_{\mathcal{B}}), \quad (2)$$

where $\text{sim}(\cdot, \cdot)$ is a hypothetical similarity measure between two sets of gradients with respect to θ_p and θ_t . θ_p and θ_t are two sets of parameters for the pretext task and the downstream task with the same network backbone. An intuitive explanation of task affinity is diagrammed in Fig. 3. However, Eq. (2) is difficult to implement in practice for two reasons. First, the training of t_p and t_d are commonly disjoint. Second, while using \mathcal{B} introduces randomness, leveraging larger amounts of data causes a non-trivial computational cost.

5.1.2. Proxy evaluation

A practical challenge in defining task affinity is that task affinity cannot be quantified directly. This is a common challenge in ML. Instead, ML researchers solve this issue by defining a *proxy* term. Following the logic in Section 5.1.1, a proxy evaluation strategy inspired by [8,9,11] is used to measure task affinity between the pretext and downstream tasks.¹ Concretely, f_θ is pre-trained by the pretext task only and frozen for later fine-tuning. In the fine-tuning phase, only g_ϕ is fine-tuned for the downstream task. The performance on the downstream task is used to reflect the task affinity between two tasks: a higher performance indicates a higher task affinity score and vice versa. In this way, f_θ and g_ϕ are optimized separately, and the downstream task will not “leak” information to f_θ . Otherwise, the downstream task performance might be dominated by the downstream task in the joint optimization process. Note, when the downstream task has fully labeled data, proxy task affinity has been a benchmark tool to compare the efficiency of pretext tasks [8,9,11].

Mathematically, let $\mathcal{V}(t_p, t_d)$ denote the downstream task performance acquired with the pretext task t_p , the downstream task t_d , and the training procedure described above. In practice, random initialization could be considered a null pretext task with zero impact on the downstream task. Thus, let $\mathcal{V}(\text{RandIni}, t_d)$ denote the downstream task performance without any pretext task, where RandIni denotes random initialization. To ensure fairness, f_θ should be randomly initialized with the same random seed for both t_p and RandIni. Thus, the *proxy task affinity* is defined as below.

Definition 1 (Proxy Task Affinity). Given the pretext task t_p and the downstream task t_d , the proxy task affinity under hyperparameters² $*$ is

$$d_{pta}(t_p, t_d | *) = \mathcal{V}(t_p, t_d) - \mathcal{V}(\text{RandIni}, t_d).$$

5.1.3. Hypotheses of proxy task affinity

With the proxy task affinity defined in Section 5.1.2, it is time to think about the role of proxy task affinity in self-supervised pre-training for partially supervised downstream tasks. Based on the empirical observations in self-supervised learning [8,9,11], self-supervised pre-training consistently improves the performance of fully supervised downstream tasks.

As pointed out in [21], tasks with high task affinity should be grouped together in the learning process to improve overall learning performance. Thus, it is natural to come up with the following hypothesis.

¹ It is worth mentioning that proxy task affinity is also called the linear classification protocol in SSL, where the downstream task is fixed to be a linear classification task with fully labeled data.

² In practice, the hyperparameters could include f_θ , g_ϕ , optimization hyperparameters, etc.. More details can be found in Section 6.

Hypothesis 1. Given a downstream task of interest, a pretext task with higher proxy task affinity leads to better downstream task performance.

The understanding of the above two hypotheses should substantially benefit the study of SSL and PSL. For the first time, the initial answers to these hypotheses are provided in Section 7 by experiments. Interestingly, both hypotheses are negated. We shall get back to this conclusion in Section 7 in detail.

5.2. Multi-pretext-task learning

Doersch et al. [23] suggests that incorporating MTL in self-supervised pre-training can *always* improve the performance under the standard supervised training setup. Dong et al. [13] has further shown that solving multiple pretext tasks can mitigate the *task misalignment* between the pretext and downstream tasks. However, the effect of simultaneously solving multiple pretext tasks on PSL remains unclear.

Hypothesis 2. Given a downstream partially supervised task, solving multiple pretext tasks can improve the downstream task performance.

Concretely, let (x^t, y^t) denote the pair of input and ground truth labels for the pretext task $t \in \mathcal{T}_{pretext}$. The optimization goal of *multi-pretext-task learning* can be defined as

$$\mathcal{L}_{MPTL} = \sum_{t \in \mathcal{T}_{pretext}} w_t \mathcal{L}_t(g_\phi^t \circ f_\theta(x^t), y^t), \quad (3)$$

where \mathcal{L}_t is the loss term for the pretext task t and w_t is the corresponding weight with $\sum_{t \in \mathcal{T}_{pretext}} w_t = 1$. Note, in Eq. (3), only f_θ is shared across pretext tasks and each pretext task has its own task-dependent prediction head.

5.3. Reverse vicinal risk minimization

[17] has shown that vicinal risk minimization (VRM) [25] can benefit the partially supervised learning on multi-object images. We further propose a pretext task based on VRM for PSMC.

5.3.1. Training

In MixUp [25], the vicinal image \tilde{x} is first generated by two randomly selected unlabeled images x_i and x_j by a linear combination, as shown in Eq. (4), where $\lambda \sim \text{Beta}(\alpha, \alpha)$ is a MixUp hyperparameter.

$$\tilde{x} = \lambda x_i + (1 - \lambda)x_j \quad (4)$$

The corresponding labels y_i and y_j are linearly added in the same way to generate a vicinal label.

$$\tilde{y} = \lambda y_i + (1 - \lambda)y_j \quad (5)$$

For VRM, the generated vicinal image and label pair are directly used in supervised learning to improve model generalization.

In contrast to VRM, we reverse the MixUp process and design the pretext task as predicting the MixUp hyperparameter λ . Let the feature extractor be f_θ and the prediction head g_ϕ^{rvrm} . The optimization goal for the pretext task is then a regression task:

$$\mathcal{L}_{VRM} = \|g_\phi^{rvrm} \circ f_\theta(x) - \lambda\|^2, \quad (6)$$

where x is simply a concatenation of (x_i, x_j, \tilde{x}) alongside the channel dimension. The pretext task is illustrated in Fig. 7.

5.3.2. Inference

In the inference phase, given an input image x_k , x is then the concatenation of (x_k, x_k, x_k) . That is to say, the input image is replicated three times. Note, in the whole training and inference process, there is no involvement of ground truth label y or vicinal label \tilde{y} . Thus, this is a self-supervised or unsupervised task. We denote this pretext task as *reverse vicinal risk minimization* (RVRM). Similar to [15], the additional computational costs caused by RVRM is trivial as Eq. (4) only involves element-wise addition supported by broadcasting. We will see later in Section 7.4 that this pretext task requires fewer computational costs than the seminal baselines while obtaining comparable performance.

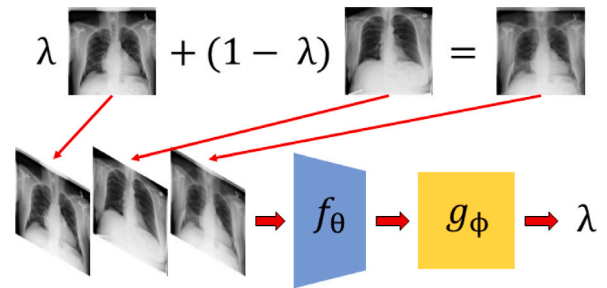


Fig. 7. Pretext task of reverse vicinal risk minimization. Given two randomly sampled images and the generated vicinal images, the network is trained to predict the MixUp hyperparameter λ .

6. Experimental setup

The purposes of the experimental design are threefold. First, we aim to understand the relationship between the pretext and downstream tasks. Second, the first empirical study of the impact of self-supervised pre-training on PSMC is provided. Third, the hypotheses proposed in Section 5 are discussed. It should be highlighted that the experiments in this section are not designed to outperform the state-of-the-art. Instead, the numerical results of the simulated experiments provide empirical insight.

6.1. Data

We use multi-label thoracic disease classification as the partially supervised downstream task. The public multi-label dataset of thoracic conditions ChestX-ray14 [19] is used. \mathcal{U} contains 10^4 unlabeled CXR images³ and \mathcal{S} contains 2200 partially labeled CXR images. For \mathcal{S} , we choose 11 common diseases among all 14 diseases, which are *infiltration, effusion, atelectasis, nodule, consolidation, pneumothorax, cardiomegaly, fibrosis, pleural thickening, mass, and emphysema*. For each disease, we randomly sample 100 positive cases and 100 negative cases as a balanced set.⁴ There is another independent test set \mathcal{T} that contains another 2200 partially labeled CXR images following the same sampling strategy as \mathcal{S} .

6.2. Implementation

Following the setup of [37], DenseNet121 [38] is chosen as the feature extractor backbone for all methods. The weighted loss is minimized by an Adam optimizer [39] with batch size 64. The learning rate is fixed to be 10^{-2} for both the pre-training and the fine-tuning stage. In the inference phase, 0.5 is set as the default threshold for the predicted probability score. Following [37], the evaluation metric is the area under the receiver operating characteristic (AUROC). For each class of interest, AUROC is computed. Then, the mean of eleven AUROCs is used to measure the overall learning performance, denoted as mAUC.

All experiments are conducted in PyTorch on an NVIDIA Tesla V100. All chest X-ray images are resized to a fixed size of 224×224 . As a pre-processing step, instance normalization [40] is performed on each chest X-ray image.

$$\hat{x}^{ij} = \frac{x^{ij} - \mu(x)}{\sigma(x)} \quad (7)$$

³ Note, \mathcal{U} follows a long-tailed distribution due to inter-class imbalance. We use this unlabeled dataset to illustrate the challenge of class imbalance in pre-training.

⁴ We aim to exclude the negative effect caused by intra-class imbalance. The other 3 diseases are too rare to create balanced sets.

In Eq. (7), x is an image, \hat{x} is the normalized image, (i, j) is the position of the pixel, and μ and σ are the mean and standard deviation of the pixels of x .

6.2.1. Proxy task affinity

We implement the proxy task affinity based on the linear classification protocol (LCP) [8,9], which has been widely used to evaluate the quality of the representations learned by SSL methods. Given a network backbone f_θ , e.g. a DenseNet121 without the last layer, the weights of f_θ are first frozen and a randomly initialized fully-connected layer (i.e. g_ϕ) is appended after f_θ as the classification head. In the training phase, only the weights of the classification head are updated. Then, the classification performance (mAUC) on the test set (i.e. \mathcal{V} in Definition 1) is used as the proxy evaluation for the learned representations.

6.2.2. Rotation prediction

The original image is rotated by $\{90 \times i\}$ degrees where i is an integer randomly sampled from a discrete uniform distribution $\{0, 1, 2, 3\}$. The auxiliary prediction head g_ϕ^{rot} is just the last layer of DenseNet121, which is a fully-connected layer with 1024 input channels and 4 output channels.

6.2.3. Masked image modeling

A 4×4 grid is first generated to split each input image into 16 patches. A patch is randomly selected out of 16 patches for each image. This patch is masked out by replacing original pixel values with zeros. The model $g_\phi^{mim} \circ f_\theta$ is implemented as a fully-convolutional network [41], where the auxiliary prediction head g_ϕ^{mim} is a 2D transposed convolutional layer with kernel size 64, stride size 32, dilation rate 1, and padding size 16.

6.2.4. Instance discrimination

The pretext task with negative cosine similarity is implemented as SimSiam.⁵ The 3-layer projector has the following architecture: $FC(1024, 1024) \rightarrow BN \rightarrow ReLU \rightarrow FC(1024, 1024) \rightarrow BN \rightarrow ReLU \rightarrow FC(1024, 2048) \rightarrow BN$, where $FC(a, b)$ denotes a fully-connected layer with a input channels and b output channels, BN denotes batch normalization [42], and ReLU denotes a rectified linear unit [43]. The 2-layer predictor has the following architecture: $FC(2048, 1024) \rightarrow BN \rightarrow ReLU \rightarrow FC(1024, 2048)$.

Previous studies on contrastive learning tend to focus on RGB images [9]. Thus, a direct application of the data augmentation policy adopted by these studies is not feasible as medical images are commonly grayscale images. To overcome this issue, we first project the image from the grayscale domain to the standard RGB domain. After data augmentation applied in the RGB domain (e.g. following [9]), we project it back to the grayscale domain.

6.2.5. Multi-pretext-task learning

As discussed in Section 5.2, the DenseNet121 backbone is shared across different pretext tasks and each pretext task has its own task-dependent prediction head. Equal weights are adopted for losses in Eq. (3).

6.2.6. Reverse vicinal risk minimization

The implementation of RVRM is based on the original implementation of MixUp.⁶ The two shape parameters of the Beta distribution are both 1 in this work.

7. Empirical analysis

7.1. Analysis on task relationships

Three pretext tasks are considered under the LCP, which are rotation prediction (ROT), masked image modeling (MIM), and instance discrimination (InstDis). As the feature extractor is fixed under the LCP, only the prediction head for the downstream task, namely PSMLC in this work, is fine-tuned. The mAUC over 11 classes is used as the proxy evaluation performance to measure the proxy task affinity between the pretext and downstream tasks.

Three pretext tasks are pre-trained for 400 epochs. The learning curves (i.e. the loss for each pretext task) are presented in Figs. 8(a)–8(c), where all losses converge. In the meantime, the corresponding learning performance under LCP is also depicted in Fig. 8(d). The reported number is the mean of mAUCs over three random seeds.

There are three important findings. First, the linear classification performance suggests that MIM has a larger proxy task affinity with PSMLC than ROT and InstDis with the current experimental setup. Second, the sign of the proxy task affinity for a pretext task and a downstream task is not fixed. ROT and InstDis can achieve lower linear classification performance than RandIni (0 epochs) with insufficient pre-training or too much pre-training. This suggests that the proxy task affinity is dependent on the number of training epochs, which can be linked with underfitting and overfitting. Last but not least, a larger number of pre-training epochs does not necessarily improve the performance, which could be dependent on the size and quality of the pre-training datasets.

7.2. Impact of self-supervised pre-training

To better understand the impact of self-supervised pre-training on PSMLC, we follow a similar experimental setup as in Section 7.1. But this time, we evaluate the performance of downstream tasks directly (c.f. LCP). That is to say, the feature extractor backbone is fine-tuned jointly with the prediction head. In this section, the two control variables are the number of partially labeled instances for each class and the number of pre-training epochs. As was done in the previous section, mAUCs under three random seeds are considered as the downstream task performance for each pretext task. The reported numbers are depicted in Fig. 9 and Fig. 10, respectively. In addition to RandIni, a randomly initialized baseline fine-tuned with full labels is included, denoted as RandIni-Full.

There are five important findings. First, a higher proxy task affinity does not always lead to a higher downstream task performance, which invalidates Hypothesis 1. A good example is MIM, which has the highest task affinity among the three pretext tasks (Fig. 8(d)). MIM tends to have lower performance than the other two pretext tasks with small n . Similarly, even with a lower task affinity than MIM, InstDis can achieve competitive or even better results. We conclude that *proxy task affinity is not a reliable tool when the downstream task is partially supervised (RQ1)*. Second, PSL baselines can even outperform the supervised learning baseline with full labels (RandIni-Full) in Fig. 10(d). This is an interesting phenomenon as a supervised learning baseline with full labels is commonly considered as *Oracle* in the literature [18], which is an upper bound for the downstream task performance. A reasonable explanation is that full labels introduce class imbalance. Note, the designed partial labels experiment has a balanced class distribution. Though full labels have ten times more class-wise labels, most of the labels belong to negative cases and common diseases have more positive cases than uncommon diseases. Third, given n , more pre-training epochs can lower the downstream task performance. This could be caused by the pre-training overfitting, i.e. the feature extractor overfits to the pretext task and exacerbates the task misalignment. Fourth, though all three pretext tasks can improve the performance of RandIni, there is no “best” pretext task. Fifth, the performance

⁵ <https://github.com/facebookresearch/simsiam>

⁶ <https://github.com/facebookresearch/mixup-cifar10>

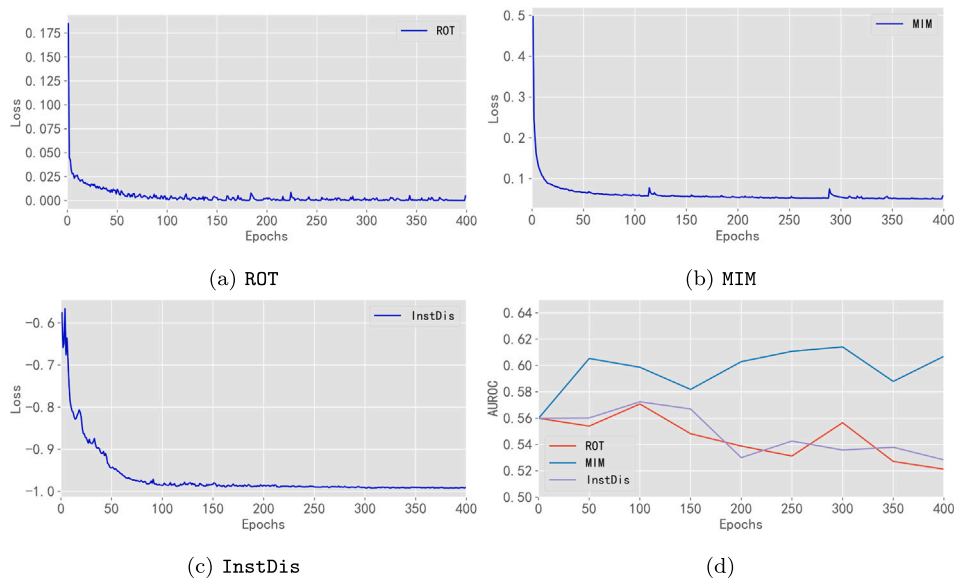


Fig. 8. (a)–(c) Learning curves for three different pretext tasks. The losses for all pretext tasks converge. (d) Linear classification performance for three different pretext tasks. The reported AUROCs are the mean mAUROCs over three random seeds. Best viewed with digital zoom.

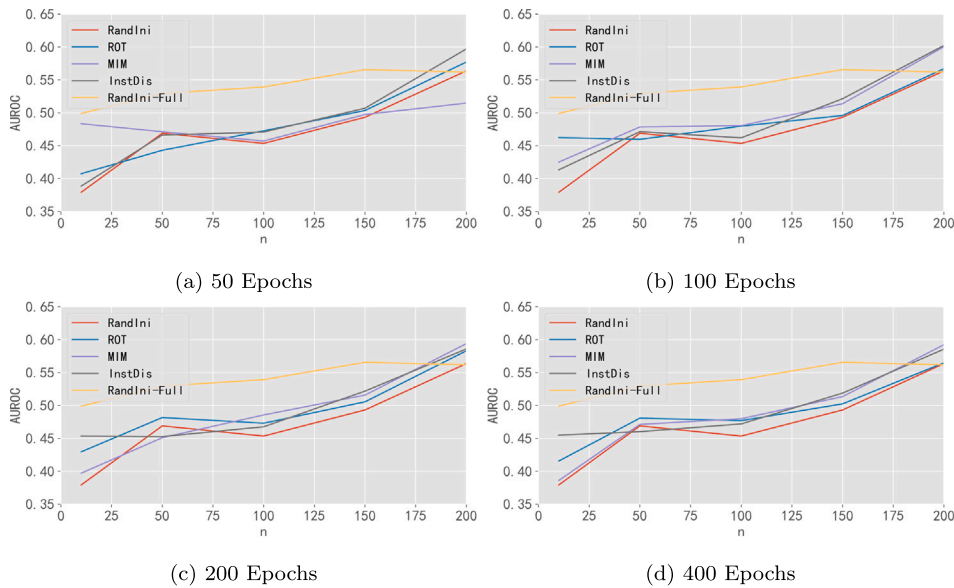


Fig. 9. Downstream task performance with different numbers of partially labeled instances for each class (n) given specific pre-training epochs. The reported AUROCs are the mean mAUROCs over three random seeds. Best viewed with digital zoom.

Table 1

Class-wise downstream task performance for pretext tasks. The reported numbers are the mean and standard deviation of AUROCs over three random seeds. Blue denotes the highest AUROC for each class (each row).

	Model					
	RandIni	ROT	MIM	InstDis	RVRM	RandIni-Full
Infiltration	0.563 ± 0.012	0.568 ± 0.017	0.597 ± 0.008	0.603 ± 0.022	0.583 ± 0.009	0.580 ± 0.054
Effusion	0.563 ± 0.017	0.577 ± 0.008	0.623 ± 0.008	0.650 ± 0.021	0.617 ± 0.021	0.632 ± 0.082
Atelectasis	0.532 ± 0.002	0.602 ± 0.028	0.602 ± 0.012	0.585 ± 0.047	0.595 ± 0.014	0.605 ± 0.043
Nodule	0.567 ± 0.031	0.583 ± 0.018	0.627 ± 0.010	0.598 ± 0.033	0.585 ± 0.015	0.508 ± 0.012
Consolidation	0.570 ± 0.011	0.632 ± 0.020	0.647 ± 0.014	0.665 ± 0.036	0.637 ± 0.035	0.548 ± 0.034
Pneumothorax	0.578 ± 0.042	0.628 ± 0.012	0.620 ± 0.012	0.595 ± 0.007	0.638 ± 0.023	0.532 ± 0.025
Cardiomegaly	0.598 ± 0.029	0.602 ± 0.006	0.625 ± 0.004	0.628 ± 0.040	0.637 ± 0.002	0.588 ± 0.035
Fibrosis	0.450 ± 0.025	0.405 ± 0.008	0.373 ± 0.008	0.438 ± 0.047	0.448 ± 0.041	0.518 ± 0.043
Pleural Thickening	0.622 ± 0.012	0.625 ± 0.016	0.632 ± 0.012	0.628 ± 0.049	0.678 ± 0.033	0.557 ± 0.018
Mass	0.543 ± 0.026	0.560 ± 0.018	0.610 ± 0.010	0.598 ± 0.022	0.593 ± 0.041	0.582 ± 0.043
Emphysema	0.613 ± 0.020	0.632 ± 0.020	0.648 ± 0.014	0.635 ± 0.047	0.615 ± 0.021	0.530 ± 0.012
Average	0.564 ± 0.016	0.583 ± 0.013	0.600 ± 0.012	0.602 ± 0.032	0.602 ± 0.016	0.562 ± 0.030

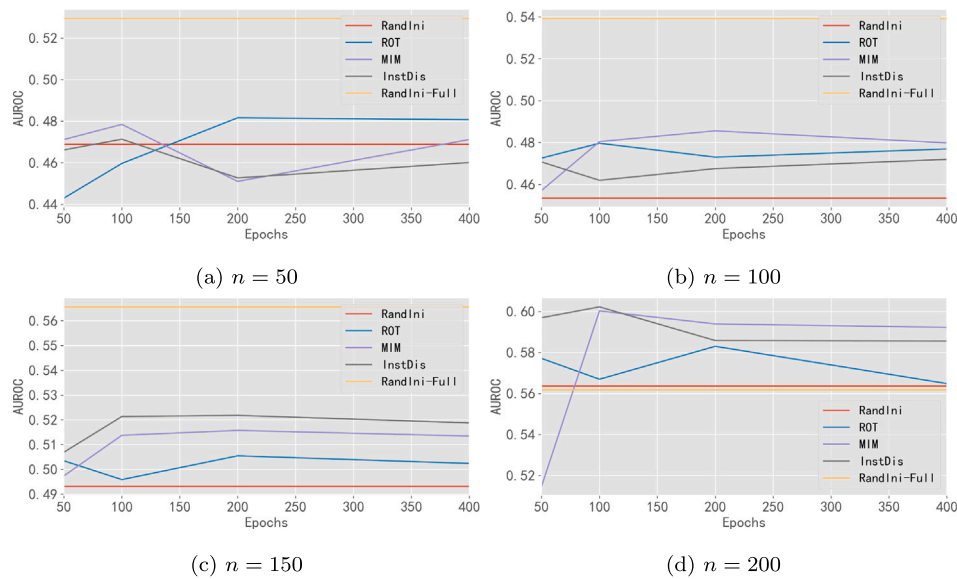


Fig. 10. Downstream task performance with different numbers of pre-training epochs given specific numbers of partially labeled instances for each class (n). The reported AUROCs are the mean mAUROCs over three random seeds. Best viewed with digital zoom.

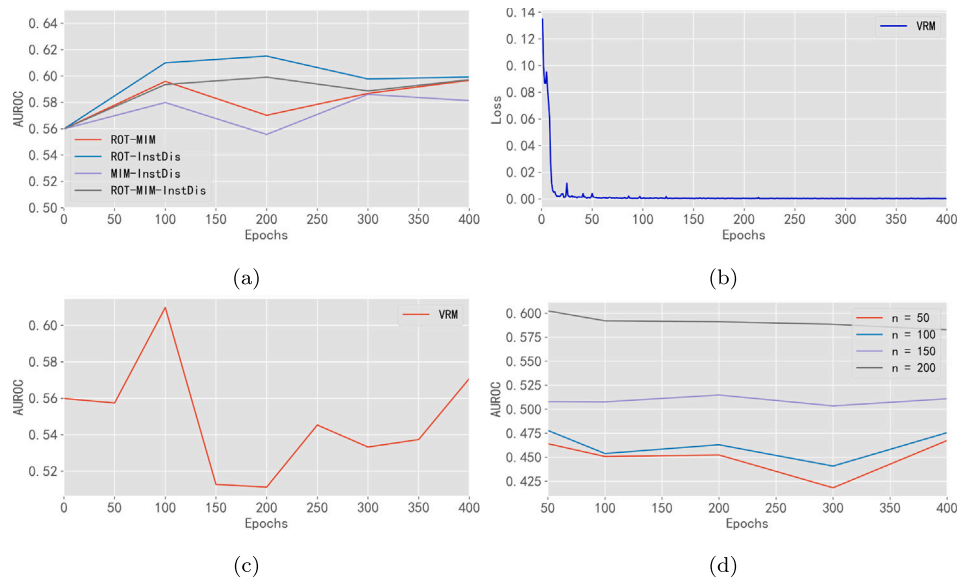


Fig. 11. (a) Linear classification performance of multi-pretex-task learning. (b) Learning curve for the pretext task based on VRM. (c) Linear classification performance of the pretext task based on VRM. (d) Downstream task performance of the pretext task based on VRM with a different number of pre-training epochs given n . The reported AUROCs are the mean mAUROCs over three random seeds. Best viewed with digital zoom.

Table 2
Class-wise downstream task performance for MPTL. The reported numbers are the mean and standard deviation of AUROCs over three random seeds. **Blue** denotes the highest AUROC for each class among MPTL baselines.

Infiltration	Model				
	ROT-MIM	ROT-INS	MIM-INS	ROT-MIM-INS	RandIni-Full
	0.607 ± 0.002	0.578 ± 0.010	0.603 ± 0.014	0.603 ± 0.008	0.580 ± 0.054
Effusion	0.657 ± 0.014	0.665 ± 0.011	0.645 ± 0.031	0.631 ± 0.006	0.632 ± 0.082
Atelectasis	0.600 ± 0.008	0.625 ± 0.004	0.588 ± 0.044	0.617 ± 0.002	0.605 ± 0.043
Nodule	0.645 ± 0.011	0.643 ± 0.005	0.615 ± 0.011	0.597 ± 0.044	0.508 ± 0.012
Consolidation	0.645 ± 0.015	0.665 ± 0.015	0.650 ± 0.011	0.643 ± 0.019	0.548 ± 0.034
Pneumothorax	0.623 ± 0.002	0.630 ± 0.029	0.630 ± 0.011	0.642 ± 0.018	0.532 ± 0.025
Cardiomegaly	0.657 ± 0.024	0.648 ± 0.006	0.650 ± 0.011	0.641 ± 0.008	0.588 ± 0.035
Fibrosis	0.373 ± 0.018	0.455 ± 0.011	0.373 ± 0.002	0.400 ± 0.025	0.518 ± 0.043
Pleural Thickening	0.650 ± 0.011	0.640 ± 0.018	0.652 ± 0.012	0.646 ± 0.013	0.557 ± 0.018
Mass	0.618 ± 0.008	0.627 ± 0.002	0.580 ± 0.004	0.613 ± 0.009	0.582 ± 0.043
Emphysema	0.663 ± 0.006	0.667 ± 0.005	0.645 ± 0.015	0.677 ± 0.006	0.530 ± 0.012
Average	0.613 ± 0.007	0.622 ± 0.001	0.603 ± 0.011	0.610 ± 0.001	0.562 ± 0.030

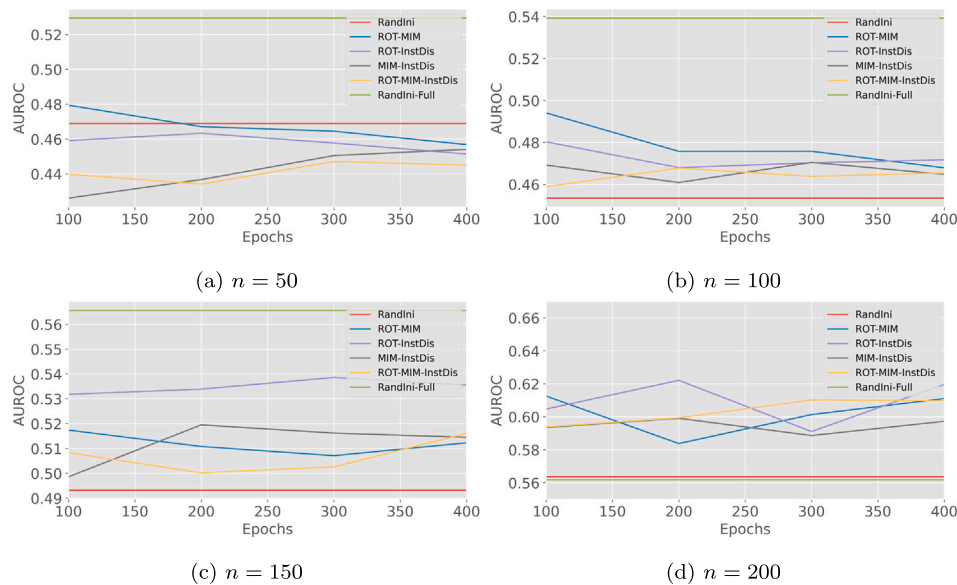


Fig. 12. Downstream task performance of multi-pretex-task learning with a different number of pre-training epochs. The reported AUROCs are the mean mAUROCs over three random seeds. Best viewed with digital zoom.

difference between different pretext tasks are diminished when labeled data for the downstream tasks get large. As shown in Fig. 9, there are clearly larger performance gaps with small n than large n for the pretext tasks. Thus, more partially labeled data can lead to diminishing returns. A similar phenomenon is also reported in the SSL literature [8] with fully labeled data.

The goal of this work is to understand the role of SSL for the downstream tasks. According to the empirical findings in SSL, the difference between the performance of different SSL baselines will be diminished when labeled data for the downstream tasks get larger. To support this claim, we use different number of partially labeled images for fine-tuning. The results are shown in Fig. 9 in the manuscript. SSL baselines clearly have larger performance gaps with small n than large n , which suggests that more partially labeled data can lead to diminishing returns.

As shown in Figs. 9 and 10, PSMLC is a challenging downstream task. The three considered pretext tasks represent three families of state-of-the-art SSL methods that succeed in multi-class classification. In contrast to multi-class classification on iconic images, multi-label classification has multiple objects at multiple locations in the same image [17]. Unfortunately, there is no existing efficient pretext task that is designed for MLC or PSMLC. As a complement to RQ1, we provide three general suggestions for choosing or designing pretext tasks. First, for downstream tasks with data scarcity, long pre-training should be avoided. An early stopping strategy can be adopted based on the learning curves. The best-performing pre-training epochs seem to overlap with the epochs when the loss starts to converge. Second, among the discussed three pretext tasks, InstDis gives the most robust performance when data scarcity is not severe in the downstream task and MIM gives the most robust performance under severe data scarcity. Third, ROT has the worse overall performance when compared with MIM and InstDis. Meanwhile, ROT also has a lower proxy task affinity and learning difficulty (easy to converge) compared to MIM and InstDis. This suggests that when designing a pretext task, the proxy task affinity and learning difficulty should be taken into consideration. The proxy task affinity under small pre-training epochs and the learning difficulty should not be low. For RQ2, though self-supervised pre-training has the potential to improve the performance of the downstream tasks, the choice of the pretext task, the number of pre-training epochs, and the number of partial labels in the fine-tuning stage are important factors to consider.

Within the downstream task (partially supervised multi-label classification), each sub-task (class) should also have its own learning difficulty. This paragraph aims to understand the impact of self-supervised pre-training on each class. Given $n = 200$, the highest downstream task performance is presented for all five baselines in Table 1. Similar to the findings in the previous section, MIM and InstDis are better choices than ROT. InstDis achieves the highest AUROCs for most classes. MIM comes second and has the lowest standard deviations most of the time. RandIni-Full only significantly outperforms MIM and InstDis on *fibrosis*, a challenging disease for PSMLC. The results in Table 1 also suggest that the performance of an MLC task can be improved by dropping a few negative labels (or weighted sampling) and transforming the problem into a PSMLC task. It is also worth mentioning that we focus more on the average performance (the bottom row in Table 1). Note, the class distribution is agnostic in the unlabeled pre-training set. [18] shows that the class distribution in the pre-training data can make a difference on the downstream task performance in a class-specific fashion. Thus, the average performance can better represent the overall performance.

7.3. Effect of multi-pretex-task learning

Following the discussion in Section 5.2, multiple pretext tasks are pre-trained together to understand the interactions between pretext tasks. As the first step, linear classification performance is reported under the LCP in Section 7.1. The results are shown in Fig. 11(a). In contrast to single pretext tasks (Fig. 8(d)), multi-pretex-task learning (MPTL) efficiently improves proxy task affinity and robustness. ROT-MIM, ROT-INS, and ROT-MIM-INS all consistently outperform the RandIni baseline (MIM-INS achieves slightly lower performance at 200 epochs), which supports Hypothesis 2, and the overfitting phenomenon with a large number of pre-training epochs seems to be efficiently alleviated.

In addition, the downstream task performance of MPTL is presented in Fig. 12, where the feature extractor and prediction head are fine-tuned jointly. MPTL does improve the performance (the highest mAUROC that a model of interest can achieve under a determined experimental setup) over single pretext tasks when enough partially labeled examples (n) are available. However, one should also notice that MPTL exposes disadvantages when n is small, e.g. ROT-MIM-INS with $n = 50$ in Fig. 12(a). Interestingly, ROT-MIM-INS does not

achieve the highest performance when compared to ROT-MIM, ROT-INS, and MIM-INS. This suggests *more pretext tasks might not always help*. Similarly, even though MIM and INS perform well alone, MIM-INS, the combination of MIM and INS, does not show the best result. The class-wise mAUCs for MPTL are summarized in Table 2 based on the best-performing epochs given $n = 200$. The empirical findings suggest that, while MPTL can improve the downstream task performance, additional attention should be paid to the size of fine-tuning dataset and choice of pretext tasks. Now, we are confident to provide an affirmative answer to RQ3. However, one should also realize the trade-off between the performance gain and additional computational cost when adopting MPTL.

Last but not least, it is worth mentioning that a brute-force grid search can find the optimal weight assignment empirically and how to find the optimal weight assignment (e.g. by evolutionary algorithm [44] or Bayesian optimization [45]) is beyond the scope of this work. Here, for simplicity, equal weights are adopted just to illustrate the effect of MPTL.

7.4. Evaluation of reverse vicinal risk minimization

We denote the proposed pretext task as RVRM. The learning curve and the linear classification performance are presented in Fig. 11(b) and Fig. 11(c), respectively. The loss for RVRM does converge and RVRM can have positive proxy task affinity. The overall downstream task performance of RVRM and the class-wise performance are presented in Fig. 11(d) and Table 1, respectively. In contrast to Fig. 10, RVRM shows competitive or even superior performance compared to other single pretext task baselines. RVRM also suffers from overfitting with long pre-training. However, one should notice that, compared with MIM and INS, the two best-performing baselines, RVRM achieves competitive performance but requires much less computational cost,⁷ e.g. RVRM does not require expensive transposed convolution operations as MIM or additional memory footprint or network architecture as INS. Meanwhile, RVRM outperforms ROT at different scales with a similar computational cost.

8. Conclusion

This work serves as the first empirical study of self-supervised pre-training for partially supervised multi-label classification and the empirical results pose a new research direction on label-efficient learning. We make a concrete step towards understanding the relationship between the pretext and downstream tasks and propose a novel pretext task reverse vicinal risk minimization which is more robust and computation-efficient than alternative seminal self-supervised tasks.

Meanwhile, we note that the current empirical study is based on multi-label classification on medical images. In the future, we will evaluate the empirical results on more challenging tasks such as object detection and segmentation with class imbalance.

CRediT authorship contribution statement

Nanqing Dong: Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Investigation, Conceptualization. **Michael Kampffmeyer:** Writing – review & editing, Validation, Supervision, Project administration. **Haoyang Su:** Writing – review & editing, Writing – original draft, Software, Methodology, Investigation. **Eric Xing:** Supervision.

⁷ Similar to [15], the computational cost of sampling and Eq. (4) is trivial in comparison with a forward and backward pass.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

- [1] V.R. de Sa, Learning classification with unlabeled data, in: NIPS, 1994, pp. 112–119.
- [2] D. Konar, S. Bhattacharyya, T.K. Gandhi, B.K. Panigrahi, A quantum-inspired self-supervised network model for automatic segmentation of brain MR images, *Appl. Soft Comput.* 93 (2020) 106348.
- [3] Á.S. Hervella, J. Rouco, J. Novo, M. Ortega, Learning the retinal anatomy from scarce annotated data using self-supervised multimodal reconstruction, *Appl. Soft Comput.* 91 (2020) 106210.
- [4] A. Elnaggar, M. Heinzinger, C. Dallago, G. Rehawi, Y. Wang, L. Jones, T. Gibbs, T. Feher, C. Angerer, M. Steinegger, et al., Prototrans: Toward understanding the language of life through self-supervised learning, *IEEE Trans. Pattern Anal. Mach. Intell.* 44 (10) (2021) 7112–7127.
- [5] Z. Tu, Z. Huang, Y. Chen, D. Kang, L. Bao, B. Yang, J. Yuan, Consistent 3d hand reconstruction in video via self-supervised learning, *IEEE Trans. Pattern Anal. Mach. Intell.* (2023).
- [6] S. Ma, J.-w. Liu, X. Zuo, Self-supervised learning for heterogeneous graph via structure information based on metapath, *Appl. Soft Comput.* 143 (2023) 110388.
- [7] F. Wang, T. Kong, R. Zhang, H. Liu, H. Li, Self-supervised learning by estimating twin class distribution, *IEEE Trans. Image Process.* (2023).
- [8] K. He, H. Fan, Y. Wu, S. Xie, R. Girshick, Momentum contrast for unsupervised visual representation learning, in: CVPR, 2020, pp. 9729–9738.
- [9] T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, in: ICML, PMLR, 2020, pp. 1597–1607.
- [10] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Pires, Z. Guo, M. Azar, et al., Bootstrap your own latent: A new approach to self-supervised learning, in: NIPS, Vol. 33, 2020, pp. 21271–21284.
- [11] X. Chen, K. He, Exploring simple siamese representation learning, in: CVPR, 2021, pp. 15750–15758.
- [12] N. Dong, M. Maggioni, Y. Yang, E. Pérez-Pellitero, A. Leonardi, S. McDonagh, Residual contrastive learning for image reconstruction: Learning transferable representations from noisy images, in: IJCAI, 2022, pp. 2930–2936.
- [13] N. Dong, M. Kampffmeyer, I. Voiculescu, Self-supervised multi-task representation learning for sequential medical images, in: ECML, Springer, 2021, pp. 779–794.
- [14] N. Dong, I. Voiculescu, Federated contrastive learning for decentralized unlabeled medical images, in: MICCAI, Springer, 2021, pp. 378–387.
- [15] N. Dong, M. Kampffmeyer, X. Liang, M. Xu, I. Voiculescu, E. Xing, Towards robust partially supervised multi-structure medical image segmentation on small-scale data, *Appl. Soft Comput.* (2022) 108074.
- [16] R. Caruana, Multitask learning, *Mach. Learn.* 28 (1) (1997) 41–75.
- [17] N. Dong, J. Wang, I. Voiculescu, Revisiting vicinal risk minimization for partially supervised multi-label classification under data scarcity, in: CVPR, 2022, pp. 4212–4220.
- [18] N. Dong, M. Kampffmeyer, I. Voiculescu, E. Xing, Federated partially supervised learning with limited decentralized medical images, *IEEE TMI* (2022).
- [19] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, R.M. Summers, Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases, in: CVPR, 2017, pp. 2097–2106.
- [20] S. Vandenhende, S. Georgoulis, B. De Brabandere, L. Van Gool, Branched multi-task networks: deciding what layers to share, in: BMVC, 2020.
- [21] C. Fifty, E. Amid, Z. Zhao, T. Yu, R. Anil, C. Finn, Efficiently identifying task groupings for multi-task learning, in: NIPS, Vol. 34, 2021, pp. 27503–27516.
- [22] A.R. Zamir, A. Sax, W. Shen, L.J. Guibas, J. Malik, S. Savarese, Taskonomy: Disentangling task transfer learning, in: CVPR, 2018, pp. 3712–3722.
- [23] C. Doersch, A. Zisserman, Multi-task self-supervised visual learning, in: ICCV, 2017, pp. 2051–2060.
- [24] O. Chapelle, J. Weston, L. Bottou, V. Vapnik, Vicinal risk minimization, in: NIPS, 2001, pp. 416–422.
- [25] H. Zhang, M. Cisse, Y.N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, in: ICLR, 2018.
- [26] Y. Zhou, Z. Li, S. Bai, C. Wang, X. Chen, M. Han, E. Fishman, A.L. Yuille, Prior-aware neural network for partially-supervised multi-organ segmentation, in: ICCV, 2019, pp. 10672–10681.
- [27] G. Shi, L. Xiao, Y. Chen, S.K. Zhou, Marginal loss and exclusion loss for partially supervised multi-organ segmentation, *Med. Image Anal.* (2021) 101979.

- [28] X. Fang, P. Yan, Multi-organ segmentation over partially labeled datasets with multi-scale feature abstraction, *IEEE TMI* 39 (11) (2020) 3619–3629.
- [29] J. Zhang, Y. Xie, Y. Xia, C. Shen, DoDNet: Learning to segment multi-organ and tumors from multiple partially labeled datasets, in: *CVPR*, 2021, pp. 1195–1204.
- [30] T. Durand, N. Mehrasa, G. Mori, Learning a deep convnet for multi-label classification with partial labels, in: *CVPR*, 2019, pp. 647–657.
- [31] M. Noroozi, P. Favaro, Unsupervised learning of visual representations by solving jigsaw puzzles, in: *ECCV*, Springer, 2016, pp. 69–84.
- [32] C. Doersch, A. Gupta, A.A. Efros, Unsupervised visual representation learning by context prediction, in: *ICCV*, 2015, pp. 1422–1430.
- [33] S. Gidaris, P. Singh, N. Komodakis, Unsupervised representation learning by predicting image rotations, in: *ICLR*, 2018.
- [34] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, A.A. Efros, Context encoders: Feature learning by inpainting, in: *CVPR*, 2016, pp. 2536–2544.
- [35] Z. Wu, Y. Xiong, S.X. Yu, D. Lin, Unsupervised feature learning via non-parametric instance discrimination, in: *CVPR*, 2018, pp. 3733–3742.
- [36] K. Wickstrøm, M. Kampffmeyer, K.Ø. Mikalsen, R. Jenssen, Mixing up contrastive learning: Self-supervised representation learning for time series, *Pattern Recognit. Lett.* 155 (2022) 54–61.
- [37] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya, et al., Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning, 2017, arXiv preprint arXiv:1711.05225.
- [38] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: *CVPR*, 2017, pp. 4700–4708.
- [39] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: *ICLR*, 2015.
- [40] W. Dai, N. Dong, Z. Wang, X. Liang, H. Zhang, E.P. Xing, SCAN: Structure correcting adversarial network for organ segmentation in chest X-rays, in: *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, Springer, 2018, pp. 263–273.
- [41] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: *CVPR*, 2015, pp. 3431–3440.
- [42] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: *ICML*, PMLR, 2015, pp. 448–456.
- [43] V. Nair, G.E. Hinton, Rectified linear units improve restricted boltzmann machines, in: *ICML*, 2010.
- [44] J. Bergstra, R. Bardenet, Y. Bengio, B. Kégl, Algorithms for hyper-parameter optimization, in: *NIPS*, 2011, pp. 2546–2554.
- [45] J. Snoek, H. Larochelle, R.P. Adams, Practical bayesian optimization of machine learning algorithms, in: *NIPS*, 2012, pp. 2951–2959.