ENHANCING INSTANCE-LEVEL IMAGE CLASSIFICATION WITH SET-LEVEL LABELS

Anonymous authors
Paper under double-blind review

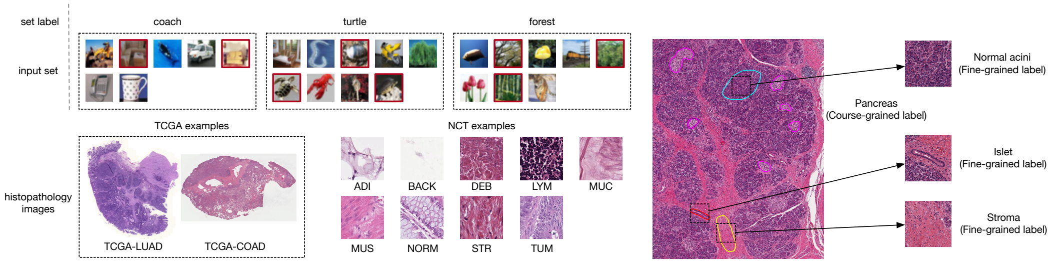
ABSTRACT

Instance-level image classification tasks have traditionally relied on single-instance labels to train models, e.g., few-shot learning and transfer learning. However, set-level coarse-grained labels that capture relationships among instances can provide richer information in real-world scenarios. In this paper, we present a novel approach to enhance instance-level image classification by leveraging set-level labels. We provide a theoretical analysis of the proposed method, including recognition conditions for fast excess risk rate, shedding light on the theoretical foundations of our approach. We conducted experiments on two distinct categories of datasets: natural image datasets and histopathology image datasets. Our experimental results demonstrate the effectiveness of our approach, showcasing improved classification performance compared to traditional single-instance label-based methods. Notably, our algorithm achieves 13% improvement in classification accuracy compared to the strongest baseline on the histopathology image classification benchmarks. Importantly, our experimental findings align with the theoretical analysis, reinforcing the robustness and reliability of our proposed method. This work bridges the gap between instance-level and set-level image classification, offering a promising avenue for advancing the capabilities of image classification models with set-level coarse-grained labels.

1 INTRODUCTION

A large amount of labeled data is typically required in traditional machine learning approaches, e.g., few-shot learning (FSL) and transfer learning (TL), to learn a robust model. However, procuring sufficient labeled data for each task is often challenging or infeasible in real-world scenarios. In this paper, we consider a novel problem setting where similar to FSL, we have a limited number of fine-grained labels in the target domain but in the source domain, we have a large amount of coarse-grained set-level labels which are easier to obtain and relevant to fine-grained labels. For example, in a digital library, there are coarse-grained set-level labels indicating the general content of photo albums, such as “beach vacation”, “nature landscapes”, or “picnic”. However, within each of these albums, there are numerous individual images, each with its own unique details and characteristics that are not explicitly labeled. In the downstream task, for instance, we care about the object classification such as “tree”, “beach”, or “mountain”. Similarly, in the medical domain, it is often useful to predict fine-grained labels of tissues, while only set-level annotations of histopathology slides are available for training at scale. We seek to enhance the downstream classification tasks with the coarse-grained set-level labels.

An effective approach to addressing the overreliance on abundant training data is FSL—a paradigm that has gained significant attention in recent years (Vinyals et al., 2016; Wang & Hebert, 2016; Triantafillou et al., 2017; Finn et al., 2017; Snell et al., 2017; Sung et al., 2018; Wang et al., 2018; Oreshkin et al., 2018; Rusu et al., 2018; Ye et al., 2018; Lee et al., 2019b; Li et al., 2019). FSL pretrains a model that can quickly adapt to new tasks using only a few labeled examples. Recent studies (Chen et al., 2019; Tian et al., 2020; Shakeri et al., 2022; Yang et al., 2022) have shown that pretraining, coupled with fine-tuning on a new task, outperforms more sophisticated episodic training methods. This involves initially training a base model on a diverse set of tasks using abundant labeled data from a source domain, and subsequently fine-tuning the model using only a small number of labeled examples specific to the target task. Despite their promising performance, existing FSL models typically depend on finely labeled source data for predicting fine-grained labels.



(a) (b)

Figure 1: (a) A collection of image sets sampled from CIFAR-100 are in the upper row. The coarse-grained label of a set is the most frequent superclass of images inside the set. WSI examples from TCGA and patches from NCT dataset are in the lower row. (b) Hierarchy of course- and fine-grained labels for histopathology images.

As an illustrative example, we consider histopathology image classification where acquiring a substantial number of fine-grained labels for individual patches (e.g., tissue labels shown in the lower row of figure 1a) is challenging. Conversely, a wealth of coarse-grained labels (e.g. the site of origin of the tumors associated with whole slide images (WSIs) from TCGA shown on the left-hand side of the upper row of figure 1a) are easily available. This motivates us to leverage these abundant coarse-grained labels and hierarchical relationships, such as between organs and tissues (as depicted in figure 1b), to enhance representation learning. Tissues consist of cellular assemblies with shared functionalities, while organs are comprised of multiple tissues. This hierarchical relationship serves as a conceptual foundation for our representation learning and provides significant contextual information for facilitating representation learning. By using coarse-grained information within this hierarchy, our goal is to learn efficiently fine-grained tissue representations within WSIs. Another example is shown in the upper row of figure 1a. We emulate a programmatic labeler that use heuristics such as keywords, regular expression, or knowledge bases to solicit sets of images. The coarse-grained labels, e.g., the most frequent superclass of images in the set, can be used to facilitate representation learning for downstream tasks such as instance-level image classification.

Our contribution In this paper, we introduce Fine-grAined representation learning from Coarse-graIned LabEls (FACILE), a novel representation learning framework that uses easily accessible coarse-grained annotations to quickly adapt to new fine-grained tasks. Distinct from existing practices in FSL, our approach utilizes coarse-grained labels in the source domain. This sets our methodology apart from conventional FSL and TL techniques, which typically rely on meticulously labeled source data to train models.

We provide an initial theoretical analysis to motivate the empirical success of FACILE and examine the convergence rate for the excess risk of downstream tasks under a novel Lipschitzness condition on the loss function concerning the fine-grained labels. Our study reveals that the availability of coarse-grained labels can lead to a substantial acceleration in the learning rate for fine-grained label prediction tasks, achieving a fast rate of $\mathcal{O}(1/n)$, where n represents the number of fine-grained data points. This analysis highlights the significant potential for leveraging coarse-grained labels to enhance the learning process in fine-grained label prediction tasks.

In our experiments, we thoroughly investigate the effectiveness of FACILE through a series of extensive experiments on natural image datasets and histopathology image datasets. For natural image datasets, we sample input sets from training data from CIFAR-100 and use the unique superclass number and most frequent superclass as coarse-grained labels. For histopathology applications, we leverage two large datasets with coarse-grained labels to pretrain our models. Subsequently, we evaluate the performance of these trained models on a diverse collection of histopathology datasets. Our algorithm achieves strong performance on 4 downstream datasets. Notably, when tested on LC25000 (Borkowski et al., 2021), our model achieves roughly 90% average ACC with 1,000 randomly sampled tasks which only have 5 fine-grained labeled data points for each of the 5 classes, outperforms the strongest baseline by roughly 6% with logistic regression fine-grained classifier. Our experiments provide compelling evidence of the efficacy and generalizability of FACILE across various datasets, highlighting its potential as a robust representation learning framework.

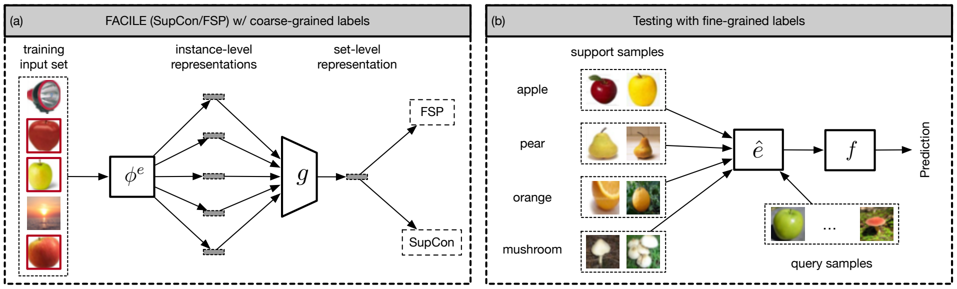


Figure 2: (a) An overview of the FACILE algorithm. The input is a set of images and the target is set-level coarse-grained label. We instantiate the $\mathcal{A}(\ell^{\text{cg}}, \mathcal{D}_m^{\text{cg}}, \mathcal{E})$ with fully supervised pretraining (FSP) with cross-entropy loss and the SupCon model. (b) Testing of few-shot learning models on fine-grained labels. The feature map $\hat{e}$ from trained models extracts instance-level features from patches of the support set and query set.

2 Fine-Grained Representation Learning from Coarse-Grained Labels

Notation Our model pretrains on a collection of samples, denoted by $\{(s_i, w_i)\}_{i=1}^m$. Each s_i is an input set of instances $\{x_j\}_{j=1}^a$, where a represents the variable input set size. $\{w_i\}$ are the set-level coarse-grained labels. The space of all instances is $\mathcal{X}$ and the space of all instance labels, which we call fine-grained labels, is $\mathcal{Y}$. The space of pretraining data is $\mathcal{S} \times \mathcal{W}$, where $\mathcal{S} = \{\{x_1, \ldots, x_a\} : x_j \in \mathcal{X} \text{ for } \forall j \in [a]\}$ and $\mathcal{W}$ denotes the space of coarse-grained labels. We receive (X, Y) from product space $\mathcal{X} \times \mathcal{Y}$ and corresponding (S, W) from product space $\mathcal{S} \times \mathcal{W}$. The goal is to predict the strong labels $y \in \mathcal{Y}$ from the instance features $x \in \mathcal{X}$.

2.1 The FACILE Algorithm

We study the model in an FSL setting where we have three datasets: (1) pretraining coarse-grained datasets $\mathcal{D}_m^{\text{cg}} = \{(s_i, w_i)\}_{i=1}^m$ sampled i.i.d. from $P_{S,W}$ (2) fine-grained support dataset $\mathcal{D}_n^{\text{fg}} = \{(x_i, y_i)\}_{i=1}^n$ sampled i.i.d., from $P_{X,Y}$, and (3) query set $\mathcal{D}^{\text{query}}$. The support set $\mathcal{D}_n^{\text{fg}}$ contains c classes and k samples x in each class (i.e., $n \equiv kc$). We assume a latent space $\mathcal{Z}$ for embedding Z. We define instance feature maps $\mathcal{E} = \{e : \mathcal{X} \to \mathcal{Z}\}$, set-input functions $\mathcal{G} = \{g : \mathcal{M} \to \mathcal{W}\}$ where $\mathcal{M} = \{\{z_1, \ldots, z_a\} : z_j \in \mathcal{Z} \text{ for } j \in [a]\}$, and fine-grained label predictors $\mathcal{F} = \{f : \mathcal{Z} \to \mathcal{Y}\}$. The corresponding set-input feature map of an instance feature map e is defined as $\phi^e : \mathcal{S} \to \mathcal{M}$. We assume the class of f is parameterized and identify f with parameter vectors for theoretical analysis. We then learn feature map e, fine-grained label predictor f, and predict fine-grained label with $f \circ e$. The schema of our model is illustrated in figure 3.

We assume two loss functions: $\ell^{\text{fg}} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ for fine-grained label prediction and $\ell^{\text{cg}} : \mathcal{W} \times \mathcal{W} \to \mathbb{R}$ for coarse-grained label prediction. ℓ^{fg} measures the loss of the fine-grained label predictor. We assume this loss is differentiable in its first argument. ℓ^{cg} measures the loss of pretraining with coarse-grained labels. For theoretical analysis, we are interested in two particular cases of ℓ^{cg}: i) $\ell^{\text{cg}}(w, w') = \mathbb{1}\{w \neq w'\}$ when $\mathcal{W}$ is a categorical space; and ii) $\ell^{\text{cg}}(w, w') = \|w - w'\|$ (for some norm $\|\cdot\|$ on $\mathcal{W}$) when $\mathcal{W}$ is a continuous space. We can also measure the loss of a feature map e by $\ell_e^{\text{cg}} = \ell^{\text{cg}}(g_e \circ \phi^e(s), w)$, where $g_e \in \arg\min_g \mathbb{E}_{P_{S,W}} \ell^{\text{cg}}(g \circ \phi^e(S), W)$. We assume there is an unknown "good" embedding $M = \phi^{e_0}(S) \in \mathcal{M}$, by which a set-input function g_{e_0} can determine W, i.e., $g_{e_0}(M) = g_{e_0} \circ \phi^{e_0}(S) = W$. The strict assumption of equality can be relaxed by incorporating an additive error term into our risk bounds of $g_{e_0} \circ \phi^{e_0}$.

Our primary goal is to learn an instance label predictor or fine-grained label predictor $\hat{f} \circ \hat{e}$ that achieves low risk $\mathbb{E}_{P_{X,Y}}[\ell^{\text{fg}}(\hat{f} \circ \hat{e}(X), Y)]$ and we can bound the excess risk:

$$\mathbb{E}_{P_{X,Y}}[\ell^{\text{fg}}(\hat{f} \circ \hat{e}(X), Y) - \ell^{\text{fg}}(f^* \circ e^*(X), Y)] \tag{1}$$

where $e^* \in \arg\min_{e \in \mathcal{E}} \mathbb{E}_{P_{S,W}} \ell_e^{\text{cg}}(S, W)$ and $f^* \in \arg\min_{f \in \mathcal{F}} \mathbb{E}_{P_{X,Y}}[\ell^{\text{fg}}(f \circ e^*(X), Y)]$.

The pseudocode for FACILE is provided in Algorithm 1, and we further illustrate the FACILE algorithm in figure 2. Given an input set s_i comprising instances $x_1, \ldots, x_a$, the feature map e is

Algorithm 1 FACILE algorithm

```
Input: loss functions $\ell^{\mathrm{fg}}$, $\ell^{\mathrm{cg}}$, predictors $\mathcal{E}$, $\mathcal{G}$, $\mathcal{F}$, datasets $\mathcal{D}_m^{\mathrm{cg}}$ and $\mathcal{D}_n^{\mathrm{fg}}$
obtain feature map $\hat{e} \leftarrow \mathcal{A}(\ell^{\mathrm{cg}}, \mathcal{D}_m^{\mathrm{cg}}, \mathcal{E})$
create dataset $\mathcal{D}_n^{\mathrm{fg,aug}} = \{(z_i, y_i) : z_i = \hat{e}(x_i), (x_i, y_i) \in \mathcal{D}_n^{\mathrm{fg}}\}_{i=1}^n$
obtain fine-grained label predictor $\hat{f} \circ \hat{e}$, where $\hat{f} \leftarrow \mathcal{A}(\ell^{\mathrm{fg}}, \mathcal{D}_n^{\mathrm{fg,aug}}, \mathcal{F})$
Return: $\hat{f} \circ \hat{e}$
```

employed to extract instance-level features for all the instances within the input set. Subsequently, a set-input model g is utilized to generate set-level features based on the instance-level features. Our FACILE model is pretrained on coarse-grained labels in conjunction with either the FSP or the SupCon model (Khosla et al., 2020). During testing, we extract the pretrained feature map $\hat{e}$ and fine-tune a classifier f using the generated embeddings from $\hat{e}$ and the fine-grained labels of the support set. The performance of the classifier $\hat{f}$ is then reported for the query set. Note that Algorithm 1 is generic since the two learning steps can use any supervised learning algorithm. Similar to Robinson et al. (2020), our analysis treats the case where $\mathcal{A}(\ell^{\mathrm{cg}}, \mathcal{D}_m^{\mathrm{cg}}, \mathcal{E})$ is empirical risk minimization (ERM) and is agnostic to the choice of $\mathcal{A}(\ell^{\mathrm{fg}}, \mathcal{D}_n^{\mathrm{fg}}, \mathcal{F})$.

2.2 Theoretical Analysis

We denote the underlying distribution of $\mathcal{D}_m^{\mathrm{cg}}$ as $P_{S,W}$ and the underlying distribution of $\mathcal{D}_n^{\mathrm{fg}}$ as $P_{X,Y}$. We assume the joint distribution of Z and Y is $P_{Z,Y}$. After we learn the feature map $\hat{e}$, we can define a new distribution $\hat{P}_{Z,Y} = P(Z,Y)\mathbb{1}\{Z = \hat{e}(X)\}$, where $\mathbb{1}$ is the indicator function. The $\mathcal{D}_n^{\mathrm{fg,aug}}$ is i.i.d. samples from $\hat{P}_{Z,Y}$. In order to include the underlying distribution of $\mathcal{D}_m^{\mathrm{cg}}$, and $\mathcal{D}_n^{\mathrm{fg}}$ into analysis, with a slight abuse of notation we use $\mathcal{A}_m(\ell^{\mathrm{cg}}, P_{S,W}, \mathcal{E})$ to denote $\mathcal{A}(\ell^{\mathrm{cg}}, \mathcal{D}_m^{\mathrm{cg}}, \mathcal{E})$ and use $\mathcal{A}_n(\ell^{\mathrm{fg}}, \hat{P}_{Z,Y}, \mathcal{F})$ to denote $\mathcal{A}(\ell^{\mathrm{fg}}, \mathcal{D}_n^{\mathrm{fg,aug}}, \mathcal{F})$.

The two learning algorithms are described as follows.

Definition 1. (Coarse-grained learning; pretraining) Let $\mathrm{Rate}_m(\ell^{\mathrm{cg}}, P_{S,W}, \mathcal{E}; \delta)$ (abbreviated to $\mathrm{Rate}_m(\ell^{\mathrm{cg}}, P_{S,W}, \mathcal{E})$) be the rate of $\mathcal{A}_m(\ell^{\mathrm{cg}}, P_{S,W}, \mathcal{E})$ which takes ℓ^{cg}, $\mathcal{E}$ and m i.i.d. observations from $P_{S,W}$ as input, and return a feature map $\hat{e} \in \mathcal{E}$ such that

$$\mathbb{E}_{P_{S,W}} \ell_{\hat{e}}^{\mathrm{cg}}(S, W) \leq \mathrm{Rate}_m(\ell^{\mathrm{cg}}, P_{S,W}, \mathcal{E}; \delta)$$

with probability at least $1 - \delta$.

Definition 2. (Fine-grained learning; downstream task learning) Let $\mathrm{Rate}_m(\ell^{\mathrm{fg}}, Q_{Z,Y}, \mathcal{F}; \delta)$ (abbreviated to $\mathrm{Rate}_m(\ell^{\mathrm{fg}}, Q_{Z,Y}, \mathcal{F})$) be the excess risk rate of $\mathcal{A}_n(\ell^{\mathrm{fg}}, Q_{Z,Y}, \mathcal{F})$ which take ℓ^{fg}, $\mathcal{F}$, and n i.i.d. observations from a distribution $Q_{Z,Y}$ as input, and returns a fine-grained predictor $\hat{f} \in \mathcal{F}$ such that $\mathbb{E}_{Q_{Z,Y}} \left[\ell_{\hat{f}}^{\mathrm{fg}}(Z, Y) - \ell_{f^*}^{\mathrm{fg}}(Z, Y) \right] \leq \mathrm{Rate}_n(\ell^{\mathrm{cg}}, Q_{Z,Y}, \mathcal{F}; \delta)$ with probability at least $1 - \delta$.

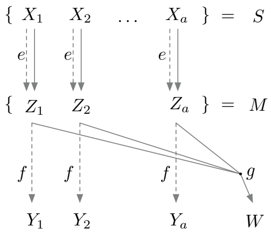

$\{ X_1 \quad X_2 \quad \dots \quad X_a \} = S$
e
e
e
$\{ Z_1 \quad Z_2 \quad Z_a \} = M$
f
f
f
g
Y_1
Y_2
Y_a
W

Figure 3: Schema of the FACILE model. The dotted lines represent the flow of fine-grained data, and the solid lines denote the flow of coarse-grained labels

Next, we introduce our relative Lipschitz assumption and the central condition for quantifying task relatedness. The Lipschitz property requires that small perturbations to the feature map e that do not harm the pretraining task, do not affect the loss of downstream task much either.

Definition 3. We say that f is L-Lipschitz relative to $\mathcal{E}$ if for all $s \in \mathcal{S}$, $x \in s$, $y \in \mathcal{Y}$, and $e, e' \in \mathcal{E}$,

$$|\ell^{\mathrm{fg}}(f \circ e(x), y) - \ell^{\mathrm{fg}}(f \circ e'(x), y)| \leq L\ell^{\mathrm{cg}}(g_e \circ \phi^e(s), g_{e'} \circ \phi^{e'}(s))$$

The function class $\mathcal{F}$ is L-Lipschitz relative to $\mathcal{E}$, if every $f \in \mathcal{F}$ is L-Lipschitz relative to $\mathcal{E}$.

Definition 3 generalizes the definition of L-Lipschitzness in Robinson et al. (2020) to bound the downstream loss deviation through the loss of the set label predictions. In the special case where $s = \{x\}$, and g is a classifier for the pretraining labels, our Lipschitz condition reduces to the Lipschitzness definition of Robinson et al. (2020).

The central condition is well-known to yield fast rates for supervised learning (Van Erven et al., 2015). Please refer to definition 6 for the definition of central condition. We show that our surrogate problem $(\ell^{\text{fg}}, \hat{P}_{Z,Y}, \mathcal{F})$ satisfies a central condition in proposition 7.

Theorem 4. *Suppose that* $(\ell^{\text{fg}}, P_{Z,Y}, \mathcal{F})$ *satisfies the central condition,* $\mathcal{F}$ *is L-Lipschitz relative to* $\mathcal{E}$*,* ℓ^{fg} *is bounded by* $B > 0$*,* $\mathcal{F}$ *is* L'*-Lipschitz in its d-dimensional parameters in the* l_2 *norm,* $\mathcal{F}$ *is contained in the Euclidean ball of radius R, and* $\mathcal{Y}$ *is compact. We also assume that* $\mathrm{Rate}_m(\ell^{\text{cg}}, P_{S,W}, \mathcal{E}) = \mathcal{O}\left(1/m^{\alpha}\right)$*. Then when* $\mathcal{A}_n(\ell^{\text{fg}}, \hat{P}_{Z,Y}, \mathcal{F})$ *is ERM we obtain excess risk* $\mathbb{E}_{P_{X,Y}}\left[\ell^{\text{fg}}_{\hat{f}\circ\hat{e}}(X,Y) - \ell^{\text{fg}}_{f^{*}\circ e^{*}}(X,Y)\right]$ *bound with probability at least* $1-\delta$ *by* $\mathcal{O}\left(\frac{d\alpha\beta\log RL'n+\log\frac{1}{\delta}}{n} + \frac{B+2L}{n^{\alpha\beta}}\right)$ *if* $m = \Omega(n^{\beta})$ *and* $\ell^{\text{cg}}(w, w') = \mathbb{1}\{w \neq w'\}$*.*

For a typical scenario where $\mathrm{Rate}_m(\ell^{\text{cg}}, P_{S,W}, \mathcal{E}) = \mathcal{O}(1/\sqrt{m})$, we can obtain fast rates with $m = \Omega(n^2)$. Similarly, in the scenario where $\mathcal{A}_m(\ell^{\text{cg}}, P_{S,W}, \mathcal{E})$ achieves fast rate, i.e., $\mathrm{Rate}_m(\ell^{\text{cg}}, P_{S,W}, \mathcal{E}) = \mathcal{O}(1/m)$, one can obtains fast rates when $m = \Omega(n)$. More generally, if $\alpha\beta \geq 1$, we observe fast rates.

We prove our theorem by first showing that the excess risk of $\hat{f} \circ \hat{e}$ can be bounded by $2L\mathrm{Rate}_m(\ell^{\text{cg}}, P_{S,W}, \mathcal{E}) + \mathrm{Rate}_n(\ell^{\text{fg}}, \hat{P}_{Z,Y}, \mathcal{F})$ in proposition 5. Then, we show that $(\ell^{\text{fg}}, \hat{P}_{Z,Y}, \mathcal{F})$ also satisfies the weak central condition in proposition 7. Thus, $\mathrm{Rate}_n(\ell^{\text{fg}}, \hat{P}_{Z,Y}, \mathcal{F})$ is also bounded by proposition 8. We refer interested readers to section H for full details of the proof.

In the next section, we first aim to empirically study the relationship between generalization error, coarse-grained dataset size, and fine-grained dataset size that our theoretical analysis predicts in section 3.3 and section 3.4. We also demonstrate the exceptional efficacy of the proposed algorithm compared to baseline models on natural image datasets and histopathology image datasets.

3 Empirical Study

3.1 Baseline Models and Algorithm Instantiation

We consider two sets of baseline models: self-supervised models (Bachman et al., 2019; He et al., 2020; Chen et al., 2020; Caron et al., 2020; Grill et al., 2020; Chen & He, 2021) and weakly supervised models (Donahue et al., 2014; Sun et al., 2017; Zeiler & Fergus, 2014; Robinson et al., 2020).

Self-supervised models Given pretraining data (S, W), self-supervised learning models ignore the labels W and learn $\hat{e}$ from S. Then, we can test $\hat{e}$ with a new task, which consists of a support set and a query set. A new model that leverages the learned $\hat{e}$ is fine-tuned on the support set and tested on the query set. We performed two self-supervised learning models in two categories, e.g., SimCLR (Chen et al., 2020) for contrastive learning and SimSiam (Chen & He, 2021) for non-contrastive learning. Details of these self-supervised learning algorithms can be found in section G.

Weakly supervised models We assign each instance, from the pretraining dataset, a label of the input set to which it belongs. We train feature map $\hat{e}$ appended with a linear classifier on the pretraining dataset. We call this model FSP-Patch, where FSP stands for fully supervised pretraining and the model is trained with the assigned instance-level labels. For a new task with a support set and a query set, we use the $\hat{e}$ to extract features for both sets, train a classifier on the support set features, and test the classifier on the query set features.

We instantiate $\mathcal{A}_m(\ell^{\text{cg}}, P_{S,W}, \mathcal{E})$ with supervised learning and the SupCon model (Khosla et al., 2020), which are denoted as FACILE-FSP and FACILE-SupCon respectively.

The ResNet18 (He et al., 2016) is used as feature maps $\mathcal{E}$. Following previous work in FSL (Tian et al., 2020; Chen et al., 2019), we use l_2-normalized features for downstream tasks. Unless otherwise specified, we evaluate methods with 1,000 randomly sampled meta-tasks from each dataset. All meta-tasks use 15 samples per class as the query set. The average F1/ACC and 95% confidence interval (CI) are reported. We follow the test setting of Yang et al. (2022) and use NearestCentroid (NC), LogisticRegression (LR), and RidgeClassifier (RC).

3.2 Pretrain with Unique Class Number of Input Sets

In order to show the advantages of using the coarse-grained labels, we introduce a new task of pretraining with the unique class number of input sets. Inspired by Lee et al. (2019a), we use the CIFAR-100 (Krizhevsky et al., 2009) dataset, which contains 100 classes grouped into 20 superclasses. We generate input sets by sampling between 6 and 10 images from CIFAR-100 training data. The targets of the input sets are the unique superclass number of the input sets.

For FACILE-FSP, we pretrain the feature map $\hat{e}$ from these input sets and targets with ℓ_1 loss. The features of CIFAR-100 test images are extracted with $\hat{e}$. Training settings of SimSiam, SimCLR and FACILE-FSP can be found in section A.1. We then test $\hat{e}$ in a few-shot manner. We random sample 5 classes, 5 examples from each class, for each meta-test dataset. The fine-grained label predictor $\hat{f}$ is trained on the support examples and tested on the query examples.

The performance of these models is reported in table 1. We can see that FACILE-FSP outperforms self-supervised learning models by a large margin.

3.3 Pretrain with Most Frequent Class Label

We sample input sets randomly from training data of CIFAR-100. The targets are the most frequent superclass of the input sets. If there is a tie in an input set, we choose a random top frequent superclass as the target of the input set. Training settings are similar to section 3.2 and can be found in section A.1. The performances of all models are reported in table 1. We can see that FACILE-FSP obtains better results compared to other models.

pretraining method	unique superclass number			most frequent superclass		
	NC	LR	RC	NC	LR	RC
SimCLR	76.07 ± 0.97	75.88 ± 1.01	75.50 ± 1.02	75.91 ± 1.00	75.82 ± 1.01	75.91 ± 1.02
SimSiam	78.15 ± 0.93	79.44 ± 0.92	79.03 ± 0.95	78.80 ± 0.93	79.44 ± 0.95	79.43 ± 0.93
FSP-Patch	N/A	N/A	N/A	73.21 ± 0.97	73.92 ± 0.98	73.40 ± 0.98
FACILE-SupCon	N/A	N/A	N/A	79.54 ± 0.92	79.54 ± 0.96	79.12 ± 0.95
FACILE-FSP	$\mathbf{86.25 \pm 0.79}$	$\mathbf{85.42 \pm 0.82}$	$\mathbf{85.84 \pm 0.81}$	$\mathbf{82.04 \pm 0.84}$	$\mathbf{81.70 \pm 0.91}$	$\mathbf{81.75 \pm 0.90}$

Table 1: Pretraining on input sets from CIFAR-100. Testing with 5-shot 5-way meta-test sets; average F1 and CI are reported.

Note that the excess risk bound of the form $b = C/n^\gamma$ implies a log-linear relationship $\log b = \log C - \gamma \log n$ between the error and the number of fine-grained labels. We can visually interpret the learning rate γ. We study two cases: when the number of coarse-grained labels m grows linearly with the number of fine-grained labels, and when the number of coarse-grained labels m grows quadratically with the number of fine-grained labels. In order to show the generalization error rate of FACILE-FSP w.r.t. fine-grained label number on CIFAR-100 test dataset, we randomly sample 5 classes (i.e., 5-way testing) for each task. We then sample $n/5$ fine-grained examples in each class for the support set and sample 15 examples for each class for the query set. The curves are shown in figure 4. The figure shows the log-linear relationship of FACILE-FSP's generalization error on downstream tasks w.r.t. fine-grained label number.

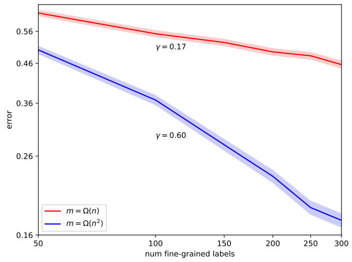


Figure 4: Generalization error (with two growth rates) of FACILE-FSP on CIFAR-100 test dataset as a function of the number of coarse-grained labels m.

3.4 Evaluation on Histopathology Images

Datasets and data extraction We pretrain our models using two independent sources of WSIs. First, we downloaded data from The Cancer Genome Atlas (TCGA) from the NCI Genomic Data Commons (GDC) (Heath et al., 2021). Two collections of non-overlapping patches with different patch sizes, i.e., 224×224 and $1,000 \times 1,000$ at 20X magnification. Background patches with high or low intensity were removed. Because the number of patches generated with size 224×224 at 20X magnification is very large, at most $1,000$ randomly selected patches are kept for each slide. The names of the tumors/organs, from which slides are collected, are used as coarse-grained labels. Second, we downloaded all clinical slides from the Genotype-Tissue Expression (GTEx) project (Lonsdale et al., 2013), which provides a resource for studying human gene expression and regulation in relation to genetic variation. We extracted non-overlapping patches with size $1,000 \times 1,000$ at 20X magnification and patches with intensity larger than 0.1 and smaller than 0.85 are kept. For these slides, we used the organs from which the tissues were extracted as coarse-grained labels. Examples and class distributions for the two datasets can be found in section C.

We test our models on 3 public datasets: LC (Borkowski et al., 2021), PAIP (Kim et al., 2021), NCT (Kather et al., 2018) and 1 private dataset PDAC. Details of these datasets are deferred to section C.

Pretrain on TCGA with patch size 224×224 We first train models on TCGA patches with size 224×224 at 20X magnification. After the models are trained, we test the feature map in these models on LC, PAIP, and NCT. Full details about FACILE-FSP, FACILE-SupCon, and baseline models' training settings can be found in section A.2.

Latent augmentation (LA) has been shown to improve FSL performance for histopathology images (Yang et al., 2022). We use faiss (Johnson et al., 2019) to perform k-means clustering. Following the setting of Yang et al. (2022), the number of prototypes in the base dictionary is 16. Each sample is augmented 100 times by LA. We refer readers to section E for details of LA.

Main results The test result is shown in table 2. In order to show the performance improvement over models pretrained on natural image datasets, we report the performance of the FSP model pretrained on ImageNet. We can see from table 2 that our model FACILE-FSP performs the best, with a large margin compared to other models. The contrastive learning model SimCLR performs worse than non-contrastive learning model SimSiam. A possible reason could be the small batch size we used for SimCLR. SimSiam maintains high performance even with small batch sizes. FSP-Patch achieves better performance compared to self-supervised learning models and the ImageNet pretrained model, which shows the usefulness of the coarse-grained labels for downstream tasks. More experiment results about test ACC on LC, PAIP and NCT datasets can be found in section B.1. Test result with larger shot number is in section B.2. We further pretrain models on GTEx and TCGA with patch size $1,000 \times 1,000$ and test the models on our private dataset PDAC. We refer readers to section B.4 for experiment results on PDAC dataset.

We show the generalization error of FACILE-FSP w.r.t. fine-grained label number in figure 5. The figure shows a clear log-linear relationship.

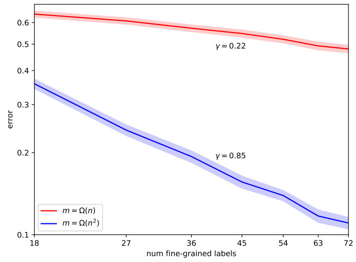


Figure 5: Generalization error on NCT dataset. The FACILE-FSP trains on TCGA dataset with m coarse-grained labels. We show the error curve with two growth rates of m.

Benefits of pretraining on Large Pathology Datasets In order to show the benefits of pretraining on large pathology datasets, we pretrain different models on the NCT training dataset and test the performance on the LC dataset, following the setting of Yang et al. (2022). Instead of separating the mixture-domain and out-domain tasks, we directly report the average F1 and CI of LR models over all 5 classes of the LC dataset. Training details of the models can be found in section B.3. The test result on the LC dataset is shown in table 3.

We can see from table 3 the best model pretrained on NCT, i.e., FSP with strong augmentation, performs worse than our model FACILE-FSP in table 2. Our method get roughly 13% improvement compared to Yang et al. (2022) on the LC dataset. The large margin between the two best models pretrained on two different datasets shows the importance of pretraining with a large number of coarse-grained labels. More results on LC and PAIP can be found in figure 7. Note that FSP with strong augmentation achieves much better performance than FSP with simple augmentation. It shows the importance of strong augmentation for pretraining. SimSiam model, trained with a batch size of 55, maintains competitive performance to MoCo v3 which needs a large batch size.

Ablation on input set size We perform an ablation study on the set size a by running FACILE-FSP with the Deep Set set-input model on the TCGA dataset. The patches in the TCGA dataset have a size of 224×224. The candidate input set sizes are 2, 5, 10, and 15. Some representative

pretraining method	NC	LR	RC	LR+LA	RC+LA
1-shot 5-way test on LC dataset					
ImageNet (FSP)	63.26 ± 1.46	63.13 ± 1.41	63.24 ± 1.40	64.51 ± 1.41	64.95 ± 1.39
SimSiam	65.83 ± 1.32	66.52 ± 1.31	66.24 ± 1.32	67.21 ± 1.29	67.83 ± 1.33
SimCLR	64.57 ± 1.36	63.85 ± 1.37	64.16 ± 1.37	65.78 ± 1.33	66.81 ± 1.40
FSP-Patch	66.73 ± 1.29	66.25 ± 1.29	66.59 ± 1.28	68.01 ± 1.24	68.28 ± 1.26
FACILE-SupCon	67.75 ± 1.31	66.11 ± 1.37	65.92 ± 1.40	68.57 ± 1.30	69.79 ± 1.33
FACILE-FSP	**75.09 ± 1.30**	**73.57 ± 1.29**	**73.16 ± 1.33**	**74.03 ± 1.28**	**72.88 ± 1.34**
5-shot 5-way test on LC dataset					
ImageNet (FSP)	82.82 ± 0.75	80.13 ± 0.82	80.23 ± 0.83	84.70 ± 0.70	84.42 ± 0.74
SimSiam	85.12 ± 0.68	82.69 ± 0.75	82.80 ± 0.76	87.45 ± 0.63	87.50 ± 0.66
SimCLR	83.45 ± 0.77	81.93 ± 0.83	81.40 ± 0.89	85.69 ± 0.73	84.93 ± 0.79
FSP-Patch	84.96 ± 0.64	84.10 ± 0.69	84.45 ± 0.68	86.31 ± 0.65	86.29 ± 0.68
FACILE-SupCon	84.77 ± 0.67	82.49 ± 0.82	82.10 ± 0.86	86.88 ± 0.64	86.92 ± 0.65
FACILE-FSP	**90.67 ± 0.54**	**89.18 ± 0.61**	**89.02 ± 0.63**	**90.03 ± 0.59**	**88.71 ± 0.67**
1-shot 3-way test on PAIP dataset					
ImageNet (FSP)	45.96 ± 1.22	47.82 ± 1.29	47.43 ± 1.29	46.38 ± 1.24	44.90 ± 1.24
SimSiam	46.43 ± 1.21	47.93 ± 1.24	47.74 ± 1.23	47.20 ± 1.21	46.31 ± 1.22
SimCLR	44.51 ± 1.16	46.44 ± 1.14	45.59 ± 1.15	45.40 ± 1.14	45.04 ± 1.16
FSP-Patch	48.85 ± 1.21	49.44 ± 1.26	50.27 ± 1.22	49.76 ± 1.20	48.44 ± 1.21
FACILE-SupCon	**52.55 ± 1.25**	**52.10 ± 1.36**	**52.40 ± 1.35**	**50.66 ± 1.28**	**50.27 ± 1.28**
FACILE-FSP	48.81 ± 1.21	50.08 ± 1.24	50.75 ± 1.23	50.03 ± 1.23	49.41 ± 1.20
5-shot 3-way test on PAIP dataset					
ImageNet (FSP)	60.73 ± 1.02	61.21 ± 1.12	61.04 ± 1.11	61.66 ± 0.91	59.30 ± 0.93
SimSiam	62.88 ± 0.97	62.59 ± 1.08	63.48 ± 1.04	65.01 ± 0.88	63.22 ± 0.89
SimCLR	60.99 ± 0.93	61.38 ± 1.00	61.62 ± 1.02	62.39 ± 0.91	61.29 ± 0.90
FSP-Patch	64.45 ± 0.92	64.60 ± 0.98	64.49 ± 0.99	64.08 ± 0.89	62.79 ± 0.89
FACILE-SupCon	65.59 ± 0.96	65.82 ± 1.06	66.72 ± 1.01	67.88 ± 0.86	66.25 ± 0.87
FACILE-FSP	**66.61 ± 0.91**	**67.57 ± 0.95**	**67.78 ± 0.95**	**68.24 ± 0.85**	**67.20 ± 0.86**
1-shot 9-way test on NCT dataset					
ImageNet (FSP)	57.35 ± 1.68	56.39 ± 1.64	56.08 ± 1.64	57.78 ± 1.66	55.85 ± 1.64
SimSiam	63.60 ± 1.62	64.43 ± 1.54	64.79 ± 1.53	65.26 ± 1.56	65.39 ± 1.53
SimCLR	59.73 ± 1.57	59.61 ± 1.57	59.34 ± 1.56	60.57 ± 1.57	60.99 ± 1.53
FSP-Patch	60.08 ± 1.46	61.55 ± 1.50	62.32 ± 1.50	61.99 ± 1.42	60.62 ± 1.38
FACILE-SupCon	64.56 ± 1.52	63.79 ± 1.53	63.94 ± 1.52	64.96 ± 1.51	65.08 ± 1.47
FACILE-FSP	**67.76 ± 1.31**	**68.52 ± 1.30**	**68.55 ± 1.28**	**68.33 ± 1.28**	**67.72 ± 1.28**
5-shot 9-way test on NCT dataset					
ImageNet (FSP)	74.59 ± 1.11	73.21 ± 1.13	74.60 ± 1.07	76.68 ± 1.04	74.39 ± 1.09
SimSiam	79.97 ± 1.05	79.81 ± 1.03	80.84 ± 0.98	83.45 ± 0.92	83.61 ± 0.90
SimCLR	76.80 ± 1.09	76.95 ± 1.07	78.25 ± 1.03	80.54 ± 0.97	81.13 ± 0.95
FSP-Patch	79.50 ± 0.94	79.54 ± 0.95	81.00 ± 0.88	82.42 ± 0.81	81.33 ± 0.79
FACILE-SupCon	79.97 ± 0.96	79.73 ± 0.98	80.58 ± 0.93	83.79 ± 0.82	84.03 ± 0.79
FACILE-FSP	**86.45 ± 0.62**	**87.74 ± 0.59**	**87.97 ± 0.58**	**88.00 ± 0.59**	**86.92 ± 0.61**

Table 2: Test result on LC, PAIP, and NCT dataset; average F1 and CI are reported.

pretraining method	NC	LR	RC	LR+LA	RC+LA
SimSiam	76.21 ± 0.87	74.05 ± 1.10	74.59 ± 1.10	77.87 ± 0.87	76.03 ± 0.94
MoCo v3 ((Yang et al., 2022))	72.82 ± 1.25	70.29 ± 1.43	71.31 ± 1.40	78.72 ± 1.00	79.71 ± 0.95
FSP (simple aug; (Yang et al., 2022))	56.44 ± 1.50	52.27 ± 1.81	55.62 ± 1.74	63.47 ± 1.37	63.47 ± 1.46
FSP (strong aug)	**83.53 ± 0.79**	**80.81 ± 1.01**	**80.27 ± 1.08**	**85.57 ± 0.77**	**84.06 ± 0.89**
SupCon	81.51 ± 0.85	78.77 ± 1.03	78.65 ± 1.08	83.51 ± 0.84	83.31 ± 0.91

Table 3: Pretraining on NCT and 5-shot 5-way testing on LC dataset; average F1 and CI are reported.

FSL methods' performance on LC dataset is shown in table 4. We can see from table 4 that there is no improvement for the LC dataset after set size $a = 5$. More ablation results on set size with LC, PAIP, and NCT dataset can be seen in appendix table 12. We refer interested readers to section F for more ablation studies about training procedures and set-input models.

set size	2	5	10	15
1-shot 5-way test				
NC	75.29 ± 1.33	**77.84 ± 1.16**	74.88 ± 1.36	75.25 ± 1.29
LR	73.72 ± 1.33	**77.56 ± 1.16**	73.84 ± 1.29	74.00 ± 1.27
LR+LA	75.27 ± 1.28	**79.16 ± 1.09**	74.41 ± 1.31	74.92 ± 1.26
5-shot 5-way test				
NC	90.62 ± 0.56	90.35 ± 0.50	90.62 ± 0.57	**90.83 ± 0.55**
LR	89.41 ± 0.63	**90.91 ± 0.47**	89.80 ± 0.59	89.63 ± 0.60
LR+LA	90.46 ± 0.58	**91.68 ± 0.50**	90.29 ± 0.57	90.46 ± 0.56

Table 4: Abation on input set size; models tested on LC dataset; average F1 and CI are reported.

4 RELATED WORK

Weakly supervised learning The concept of weakly supervised learning is introduced as a means to alleviate the annotation bottleneck in the training of machine learning models. There has been a large body of existing work in learning with only weak labels. A comprehensive survey about weakly supervised learning is provided in Zhou (2018); Zhang et al. (2022). We study a novel form of weak supervision which is provided by set-level coarse-grained labels. Among weakly supervised learning methods, Robinson et al. (2020) studied the generalization properties of weakly supervised learning and proposed a generic learning algorithm that can learn from weak and strong labels and can be proved to achieve a fast rate. The authors consider a different setting where each instance has a weak label and a strong label, and the strong label predictor learns to predict the strong labels from the instances and their corresponding embeddings learned with weak labels. We consider the setting where we have some coarse-grained labels of some sets, rather than instances and the downstream classifiers only use the learned embeddings to train and test on the downstream tasks.

Multiple-instance learning for WSIs WSI classification and regression are formulated based on multiple-instance learning (MIL) (Campanella et al., 2019; Xu et al., 2022; Ilse et al., 2018; Sharma et al., 2021; Hashimoto et al., 2020; Shao et al., 2021; Yao et al., 2020; Lu et al., 2021b;a; Chen et al., 2021b; Li et al., 2021; Chen et al., 2021a; Myronenko et al., 2021; Xiang & Zhang, 2022; Javed et al., 2022). These MIL models employ two procedures: i) feature extraction for patches cropped from a WSI and ii) aggregation of features from the same WSI. ImageNet pretrained backbones, self-supervised backbones pretrained on histopathology images, or backbones fine-tuned during training are used to extract features from patches. Deep attention pooling, graph neural networks, or sequence models, adapted for WSIs, are used for feature aggregation. In this paper, we consider a different problem setting where we enhance patch-level classification with related set-level labels. In the application of histopathology images, line 2 of our generic algorithm can be instantiated with any MIL models that have the backbones with trainable modules to extract patch-level features, e.g., Ilse et al. (2018). A complete comparison of MIL models for WSIs is out of the scope of this paper.

Learning from coarsely-labeled data Another related line of research is Phoo & Hariharan (2021), where the authors assume a taxonomy of classes with two levels, i.e., a set of fine-grained classes that are more challenging to annotate and a set of coarse-grained classes that are easier to annotate. In our paper, we do not assume a taxonomy of classes for the coarse-grained and fine-grained labels. The coarse-grained and fine-grained labels are closely related in a more sophisticated way. Besides, the inputs that are fed to models to predict the coarse-grained or fine-grained labels are different, i.e., set input for coarse-grained labels and instances for fine-grained labels.

5 CONCLUSION AND DISCUSSION

Summary We introduce FACILE, a representation learning framework that leverages coarse-grained labels for model training and enhances model performance for downstream tasks. Our theoretical analysis highlights the significant potential of leveraging set-level coarse-grained labels to benifit the learning process of fine-grained label prediction tasks. To demonstrate the effectiveness of FACILE, we conduct pretraining on CIFAR-100-based datasets and two large public histopathology datasets using coarse-grained labels and evaluate our model on a diverse collection of datasets with fine-grained labels.

Limitation and future work In this paper, we consider a novel problem setting where we enhance downstream fine-grained label classification with easily available coarse-grained labels and propose a generic algorithm that contains two supervised learning steps. As an initial step towards addressing this issue, it's important to recognize that the separate utilization of loosely related coarse-grained labels and fine-grained labels can be inefficient. Besides, the pretraining of our proposed algorithm could be expensive given the large amount of coarse-grained data and the nature of set-input models. We are motivated to explore methods for efficient and robust representation learning with coarse-grained labels. Specifically, we are interested in investigating methods of selecting a subset of the coarse-grained dataset to accelerate pretraining and make representation learning robust.

REFERENCES

Brigham & Women's Hospital & Harvard Medical School Chin Lynda 9 11 Park Peter J. 12 Kucherlapati Raju 13, Genome data analysis: Baylor College of Medicine Creighton Chad J. 22 23 Donehower Lawrence A. 22 23 24 25, Institute for Systems Biology Reynolds Sheila 31 Kreisberg Richard B. 31 Bernard Brady 31 Bressler Ryan 31 Erkkila Timo 32 Lin Jake 31 Thorsson Vesteinn 31 Zhang Wei 33 Shmulevich Ilya 31, et al. Comprehensive molecular portraits of human breast tumours. *Nature*, 490(7418):61–70, 2012.

Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *Advances in neural information processing systems*, 32, 2019.

Andrew A Borkowski, Marilyn M Bui, L Brannon Thomas, Catherine P Wilson, Lauren A DeLand, and Stephen M Mastorides. Lc25000 lung and colon histopathological image dataset, 2021.

Gabriele Campanella, Matthew G Hanna, Luke Geneslaw, Allen Miraflor, Vitor Werneck Krauss Silva, Klaus J Busam, Edi Brogi, Victor E Reuter, David S Klimstra, and Thomas J Fuchs. Clinical-grade computational pathology using weakly supervised deep learning on whole slide images. *Nature medicine*, 25(8):1301–1309, 2019.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 132–149, 2018.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

Richard J Chen, Ming Y Lu, Muhammad Shaban, Chengkuan Chen, Tiffany Y Chen, Drew FK Williamson, and Faisal Mahmood. Whole slide images are 2d point clouds: Context-aware survival prediction using patch-based graph convolutional networks. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part VIII 24*, pp. 339–349. Springer, 2021a.

Richard J Chen, Ming Y Lu, Wei-Hung Weng, Tiffany Y Chen, Drew FK Williamson, Trevor Manz, Maha Shady, and Faisal Mahmood. Multimodal co-attention transformer for survival prediction in gigapixel whole slide images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4015–4025, 2021b.

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

Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*, 2019.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15750–15758, 2021.

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*, pp. 9640–9649, 2021c.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pp. 647–655. PMLR, 2014.

Simon S Du, Wei Hu, Sham M Kakade, Jason D Lee, and Qi Lei. Few-shot learning via learning the representation, provably. *arXiv preprint arXiv:2002.09434*, 2020.

Harrison Edwards and Amos Storkey. Towards a neural statistician. *arXiv preprint arXiv:1606.02185*, 2016.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

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.

Noriaki Hashimoto, Daisuke Fukushima, Ryoichi Koga, Yusuke Takagi, Kaho Ko, Kei Kohno, Masato Nakaguro, Shigeo Nakamura, Hidekata Hontani, and Ichiro Takeuchi. Multi-scale domain-adversarial multiple-instance cnn for cancer subtype classification with unannotated histopathological images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3852–3861, 2020.

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*, pp. 770–778, 2016.

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*, pp. 9729–9738, 2020.

Allison P Heath, Vincent Ferretti, Stuti Agrawal, Maksim An, James C Angelakos, Renuka Arya, Rosita Bajari, Bilal Baqar, Justin HB Barnowski, Jeffrey Burt, et al. The nci genomic data commons. *Nature genetics*, 53(3):257–262, 2021.

Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *International conference on machine learning*, pp. 4182–4192. PMLR, 2020.

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

Syed Ashar Javed, Dinkar Juyal, Harshith Padigela, Amaro Taylor-Weiner, Limin Yu, and Aaditya Prakash. Additive mil: intrinsically interpretable multiple instance learning for pathology. *Advances in Neural Information Processing Systems*, 35:20689–20702, 2022.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

Jakob Nikolas Kather, Niels Halama, and Alexander Marx. 100,000 histological images of human colorectal cancer and healthy tissue. *https://doi.org/10.5281/zenodo.1214456*, April 2018. doi: 10.5281/zenodo.1214456. URL `https://doi.org/10.5281/zenodo.1214456`.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

Yoo Jung Kim, Hyungjoon Jang, Kyoungbun Lee, Seongkeun Park, Sung-Gyu Min, Choyeon Hong, Jeong Hwan Park, Kanggeun Lee, Jisoo Kim, Wonjae Hong, et al. Paip 2019: Liver cancer segmentation challenge. *Medical Image Analysis*, 67:101854, 2021.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *University of Toronto*, 2009.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International Conference on Machine Learning*, pp. 3744–3753. PMLR, 2019a.

Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10657–10665, 2019b.

Alexander C Li, Alexei A Efros, and Deepak Pathak. Understanding collapse in non-contrastive siamese representation learning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXI*, pp. 490–505. Springer, 2022.

Bin Li, Yin Li, and Kevin W Eliceiri. Dual-stream multiple instance learning network for whole slide image classification with self-supervised contrastive learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14318–14328, 2021.

Hongyang Li, David Eigen, Samuel Dodge, Matthew Zeiler, and Xiaogang Wang. Finding task-relevant features for few-shot learning by category traversal. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1–10, 2019.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*, 2017.

John Lonsdale, Jeffrey Thomas, Mike Salvatore, Rebecca Phillips, Edmund Lo, Saboor Shad, Richard Hasz, Gary Walters, Fernando Garcia, Nancy Young, et al. The genotype-tissue expression (gtex) project. *Nature genetics*, 45(6):580–585, 2013.

Ming Y Lu, Tiffany Y Chen, Drew FK Williamson, Melissa Zhao, Maha Shady, Jana Lipkova, and Faisal Mahmood. Ai-based pathology predicts origins for cancers of unknown primary. *Nature*, 594(7861):106–110, 2021a.

Ming Y Lu, Drew FK Williamson, Tiffany Y Chen, Richard J Chen, Matteo Barbieri, and Faisal Mahmood. Data-efficient and weakly supervised computational pathology on whole-slide images. *Nature biomedical engineering*, 5(6):555–570, 2021b.

Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6707–6717, 2020.

Andriy Myronenko, Ziyue Xu, Dong Yang, Holger R Roth, and Daguang Xu. Accounting for dependencies in deep learning based multiple instance learning for whole slide imaging. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 329–338. Springer, 2021.

Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. *Advances in neural information processing systems*, 31, 2018.

Anabik Pal, Zhiyun Xue, Kanan Desai, Adekunbiola Aina F Banjo, Clement Akinfolarin Adepiti, L Rodney Long, Mark Schiffman, and Sameer Antani. Deep multiple-instance learning for abnormal cell detection in cervical histopathology images. *Computers in Biology and Medicine*, 138: 104890, 2021.

Cheng Perng Phoo and Bharath Hariharan. Coarsely-labeled data for better few-shot transfer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9052–9061, 2021.

Colin Raffel and Daniel PW Ellis. Feed-forward networks with attention can solve some long-term memory problems. *arXiv preprint arXiv:1512.08756*, 2015.

Pierre H Richemond, Jean-Bastien Grill, Florent Altché, Corentin Tallec, Florian Strub, Andrew Brock, Samuel Smith, Soham De, Razvan Pascanu, Bilal Piot, et al. Byol works even without batch statistics. *arXiv preprint arXiv:2010.10241*, 2020.

Joshua Robinson, Stefanie Jegelka, and Suvrit Sra. Strength from weakness: Fast learning using weak supervision. In *International Conference on Machine Learning*, pp. 8127–8136. PMLR, 2020.

Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. *arXiv preprint arXiv:1807.05960*, 2018.

Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. *Advances in neural information processing systems*, 30, 2017.

Fereshteh Shakeri, Malik Boudiaf, Sina Mohammadi, Ivaxi Sheth, Mohammad Havaei, Ismail Ben Ayed, and Samira Ebrahimi Kahou. Fhist: A benchmark for few-shot classification of histological images. *arXiv preprint arXiv:2206.00092*, 2022.

Zhuchen Shao, Hao Bian, Yang Chen, Yifeng Wang, Jian Zhang, Xiangyang Ji, et al. Transmil: Transformer based correlated multiple instance learning for whole slide image classification. *Advances in neural information processing systems*, 34:2136–2147, 2021.

Yash Sharma, Aman Shrivastava, Lubaina Ehsan, Christopher A Moskaluk, Sana Syed, and Donald Brown. Cluster-to-conquer: A framework for end-to-end multi-instance learning for whole slide image classification. In *Medical Imaging with Deep Learning*, pp. 682–698. PMLR, 2021.

Baoguang Shi, Song Bai, Zhichao Zhou, and Xiang Bai. Deeppano: Deep panoramic representation for 3-d shape recognition. *IEEE Signal Processing Letters*, 22(12):2339–2343, 2015.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

Cancer Genome Atlas Research Network Tissue source sites: Duke University Medical School McLendon Roger 1 Friedman Allan 2 Bigner Darrell 1, Emory University Van Meir Erwin G. 3 4 5 Brat Daniel J. 5 6 M. Mastrogianakis Gena 3 Olson Jeffrey J. 3 4 5, Henry Ford Hospital Mikkelsen Tom 7 Lehman Norman 8, MD Anderson Cancer Center Aldape Ken 9 Alfred Yung WK 10 Bogler Oliver 11, University of California San Francisco VandenBerg Scott 12 Berger Mitchel 13 Prados Michael 13, et al. Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature*, 455(7216):1061–1068, 2008.

Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pp. 945–953, 2015.

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*, pp. 843–852, 2017.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1199–1208, 2018.

Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pp. 266–282. Springer, 2020.

Eleni Triantafillou, Richard Zemel, and Raquel Urtasun. Few-shot learning through an information retrieval lens. *Advances in neural information processing systems*, 30, 2017.

Tim Van Erven, Peter Grunwald, Nishant A Mehta, Mark Reid, Robert Williamson, et al. Fast rates in statistical and online learning. *MIT Press*, 2015.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. Order matters: Sequence to sequence for sets. *arXiv preprint arXiv:1511.06391*, 2015.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.

Yu-Xiong Wang and Martial Hebert. Learning to learn: Model regression networks for easy small sample learning. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14*, pp. 616–634. Springer, 2016.

Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7278–7286, 2018.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3733–3742, 2018.

Jinxi Xiang and Jun Zhang. Exploring low-rank property in multiple instance learning for whole slide image classification. In *The Eleventh International Conference on Learning Representations*, 2022.

Zhixin Xu, Seohoon Lim, Hong-Kyu Shin, Kwang-Hyun Uhm, Yucheng Lu, Seung-Won Jung, and Sung-Jea Ko. Risk-aware survival time prediction from whole slide pathological images. *Scientific Reports*, 12(1):21948, 2022.

Jiawei Yang, Hanbo Chen, Jiangpeng Yan, Xiaoyu Chen, and Jianhua Yao. Towards better understanding and better generalization of few-shot classification in histology images with contrastive learning. *arXiv preprint arXiv:2202.09059*, 2022.

Shuo Yang, Lu Liu, and Min Xu. Free lunch for few-shot learning: Distribution calibration. *arXiv preprint arXiv:2101.06395*, 2021.

Jiawen Yao, Xinliang Zhu, Jitendra Jonnagaddala, Nicholas Hawkins, and Junzhou Huang. Whole slide images based cancer survival prediction using attention guided deep multiple instance learning networks. *Medical Image Analysis*, 65:101789, 2020.

Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Learning embedding adaptation for few-shot learning. *arXiv preprint arXiv:1812.03664*, 7, 2018.

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

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pp. 818–833. Springer, 2014.

Jieyu Zhang, Cheng-Yu Hsieh, Yue Yu, Chao Zhang, and Alexander Ratner. A survey on programmatic weak supervision. *arXiv preprint arXiv:2202.05433*, 2022.

Renyu Zhang, Aly A Khan, and Robert L Grossman. Evaluation of hyperbolic attention in histopathology images. In *2020 IEEE 20th International Conference on Bioinformatics and Bioengineering (BIBE)*, pp. 773–776. IEEE, 2020a.

Renyu Zhang, Christopher Weber, Robert Grossman, and Aly A Khan. Evaluating and interpreting caption prediction for histopathology images. In *Machine Learning for Healthcare Conference*, pp. 418–435. PMLR, 2020b.

Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National science review*, 5(1): 44–53, 2018.

A Training Details

A.1 Pretrain with Unique Class Number and Most Frequent Class of Input Sets

SimSiam is trained for 2,000 epochs using a batch size of 512. SGD is employed with a learning rate of 0.1, weight decay of 1e-4, and momentum of 0.9. The training process incorporates a cosine scheduler.

Similarly, SimCLR is trained for 2,000 epochs with a batch size of 256. SGD is used with a learning rate of 0.05, weight decay of 1e-4, and momentum of 0.9. The training also utilizes a cosine scheduler.

FACILE-FSP is trained for 800 epochs with a batch size of 64. SGD is used with a learning rate of 0.0125, weight decay of 1e-4, and momentum of 0.9. ℓ_1 loss is optimized for pretaining with unique class numbers of input sets.

We train FSP-Patch for 800 epochs with a batch size of 256. The SGD is used with a weight decay of 1e-4, momentum of 0.9, and cosine scheduler.

For FACILE-FSP, we train the model with 2,000 epochs and a batch size of 256. The SGD is used with a learning rate of 0.05, weight decay of 1e-4, and momentum of 0.9.

A.2 Pretrain with TCGA and GTEx Dataset

In our study, an epoch refers to going through all the slides in the dataset once. In SimSiam, SimCLR, and FSP-Patch models, the data loader samples one patch for each slide. In FACILE-FSP and FACILE-SupCon, the data loader samples a set of a patches for each slide.

SimSiam is trained for 5,000 epochs using a batch size of 55. SGD is employed with a learning rate of 0.01, weight decay of 1e-4, and momentum of 0.9. The training process incorporates a cosine scheduler. Similarly, SimCLR is trained for 5,000 epochs with a batch size of 32. SGD is used with a learning rate of 0.006, weight decay of 1e-4, and momentum of 0.9. The training also utilizes a cosine scheduler.

FSP-Patch is trained for 1,000 epochs with a batch size of 64. We employ SGD with a learning rate of 0.05, weight decay of 1e-4, and momentum of 0.9. The training process includes the utilization of a cosine scheduler.

FACILE-FSP is trained for 1,000 epochs with batch size 16. The input set size is 5 by default. We employ SGD with a learning rate of 0.0125, weight decay of 1e-4, and momentum of 0.9. The training process includes the utilization of a cosine scheduler. Set Transformer with 3 inducing points and 4 attention heads is used for the set-input model g. Similarly, for our FACILE-SupCon model, we use the same input set size and set-input model. The batch size is 8. We use SGD with a learning rate of 0.0016, weight decay of 1e-4, and momentum of 0.9. We use an MLP as a projection head with two fc layers, a hidden dimension of 512, and an output dimension of 512.

B Additional Result

B.1 ACC on LC, PAIP, and NCT Datasets

We pretrain the models on TCGA datasets with patches size 224×224 at 20X magnification. Then, these pretrained models are tested on LC, PAIP, and NCT datasets. The average ACC and CI on the LC, PAIP, and NCT datasets are shown in table 5.

B.2 Test with Large Shot Number

We further test the trained models with a larger shot number k. The result is shown in table 6

pretraining method	NC	LR	RC	LR+LA	RC+LA
1-shot 5-way test on LC dataset					
ImageNet (FSP)	65.64 ± 0.49	66.06 ± 0.46	65.92 ± 0.48	66.60 ± 0.48	67.09 ± 0.47
SimSiam	68.88 ± 0.51	68.53 ± 0.48	68.27 ± 0.48	68.81 ± 0.49	70.24 ± 0.47
SimCLR	66.41 ± 0.48	66.52 ± 0.46	66.10 ± 0.46	67.70 ± 0.45	68.71 ± 0.46
FSP-Patch	68.56 ± 0.46	68.51 ± 0.45	68.68 ± 0.46	69.38 ± 0.46	69.63 ± 0.46
FACILE-SupCon	70.10 ± 0.50	69.15 ± 0.47	68.70 ± 0.46	70.35 ± 0.47	71.43 ± 0.48
FACILE-FSP	**76.76 ± 0.47**	**75.94 ± 0.44**	**75.61 ± 0.43**	**75.97 ± 0.45**	**74.76 ± 0.43**
5-shot 5-way test on LC dataset					
ImageNet (FSP)	82.79 ± 0.32	81.31 ± 0.31	81.13 ± 0.30	84.50 ± 0.30	84.73 ± 0.28
SimSiam	85.12 ± 0.30	83.39 ± 0.32	83.85 ± 0.30	87.74 ± 0.27	87.90 ± 0.26
SimCLR	83.75 ± 0.30	82.38 ± 0.30	82.32 ± 0.31	86.12 ± 0.28	85.40 ± 0.30
FSP-Patch	85.15 ± 0.29	84.38 ± 0.31	85.01 ± 0.29	86.71 ± 0.28	86.24 ± 0.27
FACILE-SupCon	85.04 ± 0.30	83.39 ± 0.29	83.44 ± 0.30	86.89 ± 0.28	87.25 ± 0.29
FACILE-FSP	**90.76 ± 0.24**	**89.67 ± 0.24**	**89.22 ± 0.25**	**90.54 ± 0.23**	**88.98 ± 0.26**
pretraining method	NC	LR	RC	LR+LA	RC+LA
1-shot 3-way test on PAIP dataset					
ImageNet (FSP)	48.44 ± 0.65	50.34 ± 0.65	50.21 ± 0.62	48.90 ± 0.62	47.51 ± 0.59
SimSiam	49.42 ± 0.65	50.25 ± 0.65	49.76 ± 0.65	49.51 ± 0.62	49.09 ± 0.63
SimCLR	47.39 ± 0.59	48.35 ± 0.59	47.97 ± 0.58	47.77 ± 0.59	47.65 ± 0.60
FSP-Patch	51.61 ± 0.68	51.61 ± 0.67	52.06 ± 0.67	51.74 ± 0.66	51.38 ± 0.66
FACILE-SupCon	**54.02 ± 0.66**	**55.51 ± 0.69**	**55.12 ± 0.69**	**54.43 ± 0.69**	**53.14 ± 0.68**
FACILE-FSP	51.77 ± 0.67	53.16 ± 0.66	52.86 ± 0.62	52.50 ± 0.68	52.49 ± 0.64
5-shot 3-way test on PAIP dataset					
ImageNet (FSP)	62.46 ± 0.52	62.48 ± 0.48	63.14 ± 0.50	62.11 ± 0.51	60.52 ± 0.49
SimSiam	63.05 ± 0.52	64.44 ± 0.49	64.66 ± 0.50	65.44 ± 0.53	64.64 ± 0.55
SimCLR	61.48 ± 0.52	61.84 ± 0.53	62.75 ± 0.51	63.03 ± 0.52	61.70 ± 0.52
FSP-Patch	65.29 ± 0.49	65.81 ± 0.51	65.98 ± 0.48	65.70 ± 0.50	64.01 ± 0.52
FACILE-SupCon	67.12 ± 0.48	67.20 ± 0.47	67.44 ± 0.50	68.18 ± 0.52	67.27 ± 0.51
FACILE-FSP	**68.23 ± 0.52**	**68.10 ± 0.51**	**68.50 ± 0.49**	**68.60 ± 0.50**	**67.81 ± 0.54**
pretraining method	NC	LR	RC	LR+LA	RC+LA
1-shot 9-way test on NCT dataset					
ImageNet (FSP)	58.75 ± 0.35	58.66 ± 0.36	58.48 ± 0.34	58.83 ± 0.36	57.32 ± 0.36
SimSiam	64.76 ± 0.40	66.09 ± 0.39	66.09 ± 0.39	66.54 ± 0.40	67.05 ± 0.41
SimCLR	60.47 ± 0.41	61.17 ± 0.38	61.43 ± 0.39	61.65 ± 0.40	62.48 ± 0.38
FSP-Patch	61.03 ± 0.42	63.53 ± 0.40	63.26 ± 0.42	62.75 ± 0.43	61.57 ± 0.42
FACILE-SupCon	65.63 ± 0.39	65.46 ± 0.38	65.56 ± 0.38	66.24 ± 0.38	65.60 ± 0.41
FACILE-FSP	**69.24 ± 0.44**	**70.32 ± 0.43**	**70.36 ± 0.41**	**69.56 ± 0.41**	**69.06 ± 0.42**
5-shot 9-way test on NCT dataset					
ImageNet (FSP)	74.82 ± 0.26	74.35 ± 0.26	75.20 ± 0.26	77.11 ± 0.23	74.89 ± 0.26
SimSiam	80.59 ± 0.23	80.51 ± 0.23	81.54 ± 0.21	83.68 ± 0.22	83.85 ± 0.22
SimCLR	77.30 ± 0.25	77.64 ± 0.24	79.17 ± 0.24	80.99 ± 0.24	81.71 ± 0.23
FSP-Patch	79.61 ± 0.25	79.89 ± 0.24	81.71 ± 0.23	82.92 ± 0.24	81.67 ± 0.24
FACILE-SupCon	80.14 ± 0.25	80.11 ± 0.22	81.41 ± 0.24	84.12 ± 0.22	84.27 ± 0.22
FACILE-FSP	**86.87 ± 0.22**	**87.70 ± 0.20**	**88.53 ± 0.19**	**88.21 ± 0.20**	**87.24 ± 0.22**

Table 5: Models tested on LC, PAIP, and NCT dataset; average ACC and CI are reported.

B.3 Benefits of Pretraining on Large Pathology Datasets

In order to demonstrate the advantages of pretraining on large pathology datasets, we compare the performance of models pretrained on TCGA datasets with those pretrained on NCT datast, which are also studied in Yang et al. (2022).

The SimSiam model is trained for 100 epochs. SGD optimizer is used with learning rate of 0.01, weight decay of 0.0001, momentum of 0.9, and cosine learning rate decay. The batch size is 55.

For MoCo v3, similar to (Chen et al., 2021c; Yang et al., 2022), LARS optimizer (You et al., 2017) was used with an initial learning rate of 0.3, weight decay of $1.5e - 6$, the momentum of 0.9, and cosine decay schedule. MoCo v3 was trained with a batch size of 256 for 200 epochs.

The FSP model with simple augmentation follows the setting of Yang et al. (2022). SGD optimizer with learning rate of 0.5, momentum of 0.9 and weight decay of 0 are used. A large batch size is used 512. The model is trained for 100 epochs with "step decay" schedule. The learning rate multiplied by 0.1 at 30, 60, and 90 epochs respectively. The FSP model with strong augmentation was trained for 50 epochs. The batch size is set to 64. The SGD is used with a learning rate of 0.03,

pretraining method	NC	LR	RC	LR+LA	RC+LA
10-shot 5-way on LC					
ImageNet (FSP)	78.76 ± 0.94	78.92 ± 0.92	80.45 ± 0.87	82.25 ± 0.83	80.20 ± 0.89
SimSiam	88.52 ± 0.55	87.20 ± 0.58	87.73 ± 0.56	91.62 ± 0.46	91.88 ± 0.47
SimCLR	87.02 ± 0.64	86.26 ± 0.64	85.61 ± 0.72	90.28 ± 0.52	89.60 ± 0.58
FSP-Patch	88.41 ± 0.53	88.64 ± 0.52	89.15 ± 0.51	90.49 ± 0.50	89.88 ± 0.54
FACILE-SupCon	87.58 ± 0.56	87.00 ± 0.61	87.16 ± 0.63	90.75 ± 0.49	91.02 ± 0.49
FACILE-FSP	**92.81 ± 0.42**	**92.33 ± 0.44**	**92.44 ± 0.43**	**93.07 ± 0.43**	**92.35 ± 0.48**
10-shot 3-way on PAIP					
ImageNet (FSP)	65.36 ± 0.91	65.17 ± 1.00	65.40 ± 0.99	66.52 ± 0.81	64.45 ± 0.81
SimSiam	67.19 ± 0.88	67.35 ± 0.98	68.55 ± 0.94	70.88 ± 0.77	70.62 ± 0.77
SimCLR	65.77 ± 0.85	66.70 ± 0.91	67.01 ± 0.91	68.41 ± 0.79	66.96 ± 0.82
FSP-Patch	68.50 ± 0.82	69.12 ± 0.85	69.39 ± 0.85	70.13 ± 0.75	68.25 ± 0.76
FACILE-SupCon	69.79 ± 0.85	69.78 ± 0.90	70.36 ± 0.88	73.53 ± 0.73	72.53 ± 0.73
FACILE-FSP	**71.93 ± 0.82**	**73.02 ± 0.84**	**73.54 ± 0.80**	**74.19 ± 0.71**	**72.70 ± 0.72**
10-shot 9-way on NCT					
ImageNet (FSP)	78.76 ± 0.94	78.92 ± 0.92	80.45 ± 0.87	82.25 ± 0.83	80.20 ± 0.89
SimSiam	82.92 ± 0.91	83.42 ± 0.89	84.76 ± 0.81	87.66 ± 0.72	88.12 ± 0.69
SimCLR	80.34 ± 0.96	81.67 ± 0.90	83.09 ± 0.84	85.96 ± 0.76	86.82 ± 0.72
FSP-Patch	83.36 ± 0.77	84.05 ± 0.74	85.93 ± 0.65	87.15 ± 0.62	86.05 ± 0.63
FACILE-SupCon	83.05 ± 0.83	83.84 ± 0.80	84.95 ± 0.73	88.29 ± 0.60	88.58 ± 0.61
FACILE-FSP	**89.55 ± 0.50**	**90.57 ± 0.47**	**91.28 ± 0.45**	**91.43 ± 0.45**	**90.77 ± 0.47**

Table 6: Test result on LC, PAIP, and NCT dataset with shot number 10; average F1 and CI are reported.

momentum of 0.9, weight decay of 0.0001, and the cosine schedule. The model is trained for 50 epochs.

The SupCon model is trained with trained for 100 epochs. The batch size is set to 64. The SGD optimizer is used with a learning rate of 0.01, momentum of 0.9, weight decay of 0.0001, and the cosine schedule.

table 7 shows the performance of the pretrained models on the LC and PAIP dataset with shot numbers 1 or 5. Notably, the best-performing models on the two test datasets exhibit a significant performance gap compared to the best models pretrained on TCGA datasets as depicted in table 2.

pretraining method	NC	LR	RC	LR+LA	RC+LA
1-shot 5-way test on LC dataset					
SimSiam	59.30 ± 1.31	58.67 ± 1.41	58.58 ± 1.40	59.66 ± 1.35	59.85 ± 1.35
MoCo v3 ((Yang et al., 2022))	59.38 ± 1.62	59.39 ± 1.68	59.46 ± 1.68	60.15 ± 1.59	60.54 ± 1.58
FSP (simple aug; (Yang et al., 2022))	51.42 ± 1.59	46.06 ± 1.88	46.33 ± 1.86	50.53 ± 1.65	51.00 ± 1.65
FSP (strong aug)	**68.00 ± 1.29**	**66.17 ± 1.41**	**66.18 ± 1.46**	**68.39 ± 1.34**	**68.02 ± 1.40**
SupCon	64.48 ± 1.33	63.52 ± 1.42	63.84 ± 1.40	65.43 ± 1.33	65.98 ± 1.38
5-shot 5-way test on LC dataset					
SimSiam	76.21 ± 0.87	74.05 ± 1.10	74.59 ± 1.10	77.87 ± 0.87	76.03 ± 0.94
MoCo v3 ((Yang et al., 2022))	72.82 ± 1.25	70.29 ± 1.43	71.31 ± 1.40	78.72 ± 1.00	79.71 ± 0.95
FSP (simple aug; (Yang et al., 2022))	56.44 ± 1.50	52.27 ± 1.81	55.62 ± 1.74	63.47 ± 1.37	63.47 ± 1.46
FSP (strong aug)	**83.53 ± 0.79**	**80.81 ± 1.01**	**80.27 ± 1.08**	**85.57 ± 0.77**	**84.06 ± 0.89**
SupCon	81.51 ± 0.85	78.77 ± 1.03	78.65 ± 1.08	83.51 ± 0.84	83.31 ± 0.91
1-shot 3-way test on PAIP dataset					
SimSiam	37.13 ± 1.14	38.26 ± 1.13	37.93 ± 1.15	38.00 ± 1.12	38.67 ± 1.12
MoCo v3 ((Yang et al., 2022))	43.17 ± 1.26	42.48 ± 1.30	43.02 ± 1.31	43.55 ± 1.28	44.57 ± 1.28
FSP (simple aug; (Yang et al., 2022))	37.15 ± 1.07	36.69 ± 1.13	37.39 ± 1.08	37.40 ± 1.07	35.28 ± 1.09
FSP (strong aug)	47.67 ± 1.18	48.44 ± 1.19	48.16 ± 1.21	48.27 ± 1.17	**49.38 ± 1.19**
SupCon	**48.45 ± 1.19**	**49.29 ± 1.20**	**48.97 ± 1.22**	**49.47 ± 1.20**	48.53 ± 1.20
5-shot 3-way test on PAIP dataset					
SimSiam	47.52 ± 1.00	48.12 ± 1.10	47.04 ± 1.11	52.70 ± 0.95	54.51 ± 1.00
MoCo v3 ((Yang et al., 2022))	55.43 ± 1.00	54.23 ± 1.09	54.05 ± 1.09	56.07 ± 0.92	55.73 ± 0.93
FSP (simple aug; (Yang et al., 2022))	44.98 ± 0.95	45.13 ± 0.96	45.30 ± 0.96	44.34 ± 0.87	44.03 ± 0.88
FSP (strong aug)	62.00 ± 0.88	62.48 ± 0.97	62.04 ± 0.98	**64.82 ± 0.86**	**64.60 ± 0.87**
SupCon	**63.62 ± 0.91**	**64.38 ± 0.96**	**63.61 ± 1.00**	64.37 ± 0.87	64.28 ± 0.88

Table 7: pretraining on NCT dataset and testing on LC and PAIP dataset; average F1 and CI are reported.

B.4 PRETRAIN ON TCGA AND GTEX WITH PATCH SIZE 1,000X1,000

We train the models using TCGA patches of size $1,000 \times 1,000$, which are extracted from 20X magnification and resized to 224×224. Subsequently, the pretrained models are evaluated on PDAC datasets, and the corresponding test performance is presented in Figure 8. Notably, for shot number of 1 and 5, our model significantly outperforms other models, demonstrating a substantial performance margin.

pretraining method	NC	LR	RC	LR+LA	RC+LA
1-shot 5-way test					
ImageNet (FSP)	29.57 ± 1.07	31.32 ± 1.09	31.16 ± 1.07	30.88 ± 1.08	30.14 ± 1.08
SimSiam	30.48 ± 1.08	30.18 ± 1.12	30.19 ± 1.13	30.41 ± 1.08	31.13 ± 1.10
SimCLR	30.79 ± 1.08	30.93 ± 1.13	30.78 ± 1.12	31.33 ± 1.08	31.22 ± 1.07
FSP-Patch	34.04 ± 1.16	33.99 ± 1.20	33.69 ± 1.20	34.29 ± 1.15	34.99 ± 1.16
FACILE-SupCon	33.72 ± 1.18	32.13 ± 1.26	31.93 ± 1.26	34.54 ± 1.17	34.30 ± 1.20
FACILE-FSP	$\mathbf{37.36 \pm 1.16}$	$\mathbf{36.07 \pm 1.23}$	$\mathbf{36.93 \pm 1.21}$	$\mathbf{36.79 \pm 1.19}$	$\mathbf{36.81 \pm 1.18}$
5-shot 5-way test					
ImageNet (FSP)	41.83 ± 0.96	41.30 ± 1.10	41.08 ± 1.08	42.38 ± 0.94	41.29 ± 0.93
SimSiam	40.15 ± 1.03	37.29 ± 1.21	37.43 ± 1.21	41.87 ± 1.00	42.70 ± 1.01
SimCLR	40.30 ± 1.04	38.74 ± 1.19	39.02 ± 1.16	40.98 ± 0.96	40.90 ± 0.98
FSP-Patch	44.26 ± 1.10	42.99 ± 1.20	43.69 ± 1.12	46.32 ± 0.97	46.69 ± 0.96
FACILE-SupCon	42.51 ± 1.12	40.27 ± 1.25	40.75 ± 1.24	45.10 ± 0.99	45.73 ± 0.99
FACILE-FSP	$\mathbf{48.21 \pm 1.04}$	$\mathbf{47.62 \pm 1.12}$	$\mathbf{47.94 \pm 1.08}$	$\mathbf{48.84 \pm 0.95}$	$\mathbf{48.37 \pm 0.95}$

Table 8: Models pretrained on TCGA and tested on PDAC dataset; average F1 and CI are reported.

Similarly, we train the models using GTEx patches with dimensions of $1,000 \times 1,000$. The patches are extracted from 20X magnification and resized to 224×224. The pretrained models are tested on PDAC datasets, revealing similar outcomes, as illustrated in Table 9.

pretraining method	NC	LR	RC	LR+LA	RC+LA
1-shot 5-way test					
SimSiam	34.78 ± 1.18	34.57 ± 1.25	35.30 ± 1.25	35.13 ± 1.19	35.27 ± 1.19
SimCLR	33.68 ± 1.14	33.74 ± 1.18	33.69 ± 1.17	34.28 ± 1.14	33.84 ± 1.12
FSP-Patch	31.87 ± 1.09	32.90 ± 1.13	32.53 ± 1.11	32.55 ± 1.09	32.10 ± 1.07
FACILE-SupCon	34.36 ± 1.06	34.35 ± 1.13	34.39 ± 1.14	34.70 ± 1.07	34.35 ± 1.07
FACILE-FSP	$\mathbf{35.62 \pm 1.10}$	$\mathbf{35.51 \pm 1.15}$	$\mathbf{35.40 \pm 1.13}$	$\mathbf{35.87 \pm 1.10}$	$\mathbf{36.16 \pm 1.09}$
5-shot 5-way test					
SimSiam	46.00 ± 1.10	43.26 ± 1.30	44.19 ± 1.26	47.24 ± 1.00	$\mathbf{47.85 \pm 1.00}$
SimCLR	44.44 ± 1.08	43.40 ± 1.19	43.58 ± 1.15	44.60 ± 0.98	44.17 ± 0.96
FSP-Patch	42.09 ± 0.99	40.15 ± 1.15	40.69 ± 1.09	42.71 ± 0.92	42.66 ± 0.90
FACILE-SupCon	44.85 ± 1.02	43.65 ± 1.15	44.01 ± 1.13	46.37 ± 0.93	45.10 ± 0.92
FACILE-FSP	$\mathbf{46.91 \pm 0.97}$	$\mathbf{46.32 \pm 1.07}$	$\mathbf{47.10 \pm 1.02}$	$\mathbf{48.01 \pm 0.90}$	47.70 ± 0.89

Table 9: Models pretrained on GTEx and tested on PDAC dataset; average F1 and CI are reported.

C DATASETS

C.1 GTEX DATASET

The Genotype-Tissue Expression (GTEx) project is a pioneering initiative aimed at constructing an extensive public repository to investigate tissue-specific gene expression and regulation. The GTEx project collected samples from 54 non-diseased tissue sites across nearly 1000 individuals, with an emphasis on molecular assays such as Whole Genome Sequencing (WGS), Whole Exome Sequencing (WES), and RNA-sequencing. Additionally, the GTEx Biobank contains a plethora of unutilized samples. The GTEx portal (`https://gtexportal.org/home/`) provides unrestricted access to a plethora of data, including gene expression levels, quantitative trait loci (QTLs), and histology images, to aid the research community in advancing our understanding of human gene expression and its regulation.

We downloaded all the slides from the GTEx portal. The organs from which the slides are extracted are used for coarse-grained labels. We extract all the non-overlapping patches with size $1,000 \times 1,000$ and only keep those with intensity in $[0.1, 0.85]$ to filter out backgrounds.

The number of slides from each organ for GTEx can be found in figure 6. Thumbnails of WSI examples from the GTEx dataset can be found in figure 7.

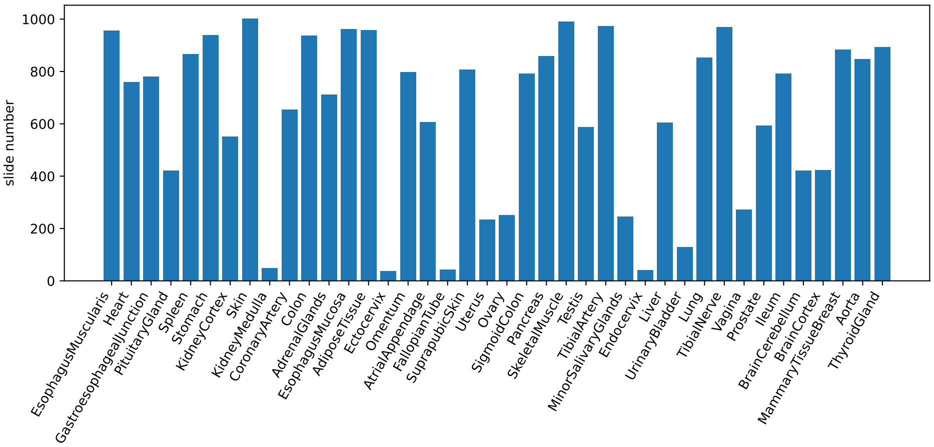


Figure 6: Slide number for each organ in GTEx

C.2 TCGA Dataset

The Cancer Genome Atlas (TCGA; `https://www.cancer.gov/ccg/research/genome-sequencing/tcga`) is a project that aims to comprehensively characterize genetic mutations responsible for cancer using genome sequencing and bioinformatics. The TCGA dataset consists of 10,825 patient samples, including gene expression, DNA methylation, copy number variation, and mutation data, histopathology data, among others (source sites: Duke University Medical School McLendon Roger 1 Friedman Allan 2 Bigner Darrell 1 et al., 2008; 13 et al., 2012). This large-scale dataset has enabled researchers to identify numerous genomic alterations associated with cancer and has contributed to the development of new diagnostic and therapeutic approaches.

We downloaded all the diagnostic slides from GDC portal `https://portal.gdc.cancer.gov/`. The project names of the slides are used for coarse-grained labels. We extract patches at two different scales, i.e., 224×224 and $1,000 \times 1,000$ at 20X magnification, from all the slides.

The number of slides from each project for TCGA can be found in figure 8. Thumbnails of WSI examples from TCGA dataset can be found in figure 9.

C.3 PDAC Dataset

To address the presence of multiple tissues within certain patches, we employ a labeling strategy that involves identifying and labeling the centered tissues within these patches. To ensure annotation accuracy, each patch undergoes labeling by a minimum of two pathologists, thereby maintaining the quality of the annotations. For the specific patch numbers corresponding to each tissue in the PDAC dataset, please refer to Figure 10. Furthermore, examples of patches from the PDAC dataset are provided in Figure 11, offering visual illustrations of the dataset.

In order to validate our model on a real-world dataset, we generated WSIs of Pancreatic Ductal Adenocarcinoma (PDAC)[1]. PDAC, a particularly aggressive and lethal form of cancer originating in the pancreatic duct cells, presents various subtypes, each with distinct morphological characteristics. These variations underscore the need for advanced automated tools to accurately characterize and differentiate between these subtypes, thereby aiding disease studies and potentially informing treatment strategies. Examples of PDAC and class distribution are detailed in section C. There are in total 12,250 annotated patches extracted from 194 slides. The patch size used for this analysis is

[1] We will make data publicly available upon acceptance of our paper

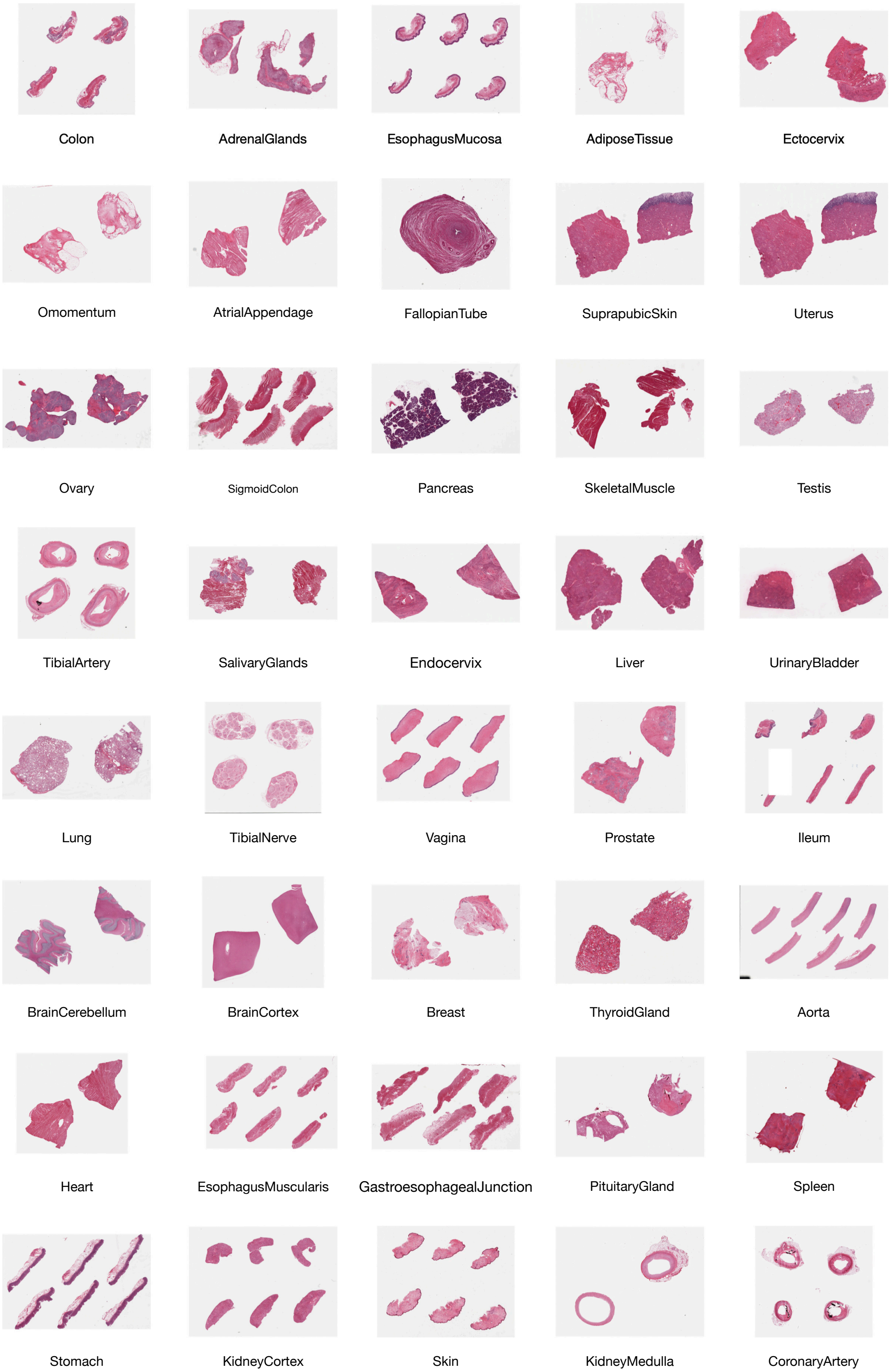


Figure 7: Randomly deleted examples from GTEx dataset

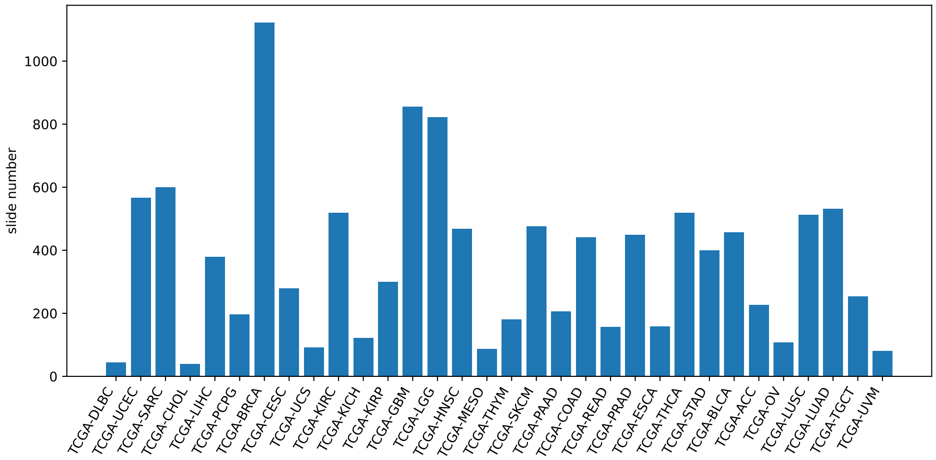


Figure 8: Slide number for each tumor in TCGA

Coarse-Grained Dataset	Data type and annotation	WSI number	Extracted patch number
GTEx	slides; organs	25,501	9,465,689
TCGA	slides; tumors	11,638	10,321,273 (11,588,226 w/ size 224)
Fine-Grained Dataset	Data type and annotation	WSI number	Extracted patch number
PDAC	patches; tissues	194	12,250
LC25000	patches; tissues	1,250	25,000
PAIP19	patches; tissues	60	75,000
NCT-CRC-HE-100K	patches; tissues	86	100,000

Table 10: Dataset statistics

$1,000 \times 1,000$ at a 20X magnification. Each patch was annotated into one of 5 classes (i.e., Stroma, Normal Acini, Normal Duct, Tumor, and Islet) and confirmed by at least two pathologists.

C.4 NCT, PAIP, and LC

We test our models on 4 datasets with fine-grained labels. These datasets are from diverse body sites. Statistics of these datasets can be found in table 10.

NCT-CRC-HE-100K (NCT) is collected from colon (Kather et al., 2018). It consists of 9 classes with 100K non-overlapping patches. The patch size is 224 × 224. LC25000 (LC) is collected from lung and colon sites (Borkowski et al., 2021). It has 5 classes and each class has 5,000 patches. The patch size is 768 × 768. We resize the patches to 224 × 224. PAIP19 (PAIP) is collected from liver site (Kim et al., 2021). There are in total 50 WSIs. The WSIs are cropped into patches with size 224 × 224. We only keep those patches with masks and assign labels with majority voting similar to Yang et al. (2022). We downsample these patches to 75K patches, with 25K in each class.

D Data Augmentation

Two data augmentation strategies are used in this paper.

Simple augmentation Following Yang et al. (2022), we also used a simple augmentation policy which includes random resized cropping and horizontal flipping. In our paper, this simple augmentation policy is only used for SupCE model pretraining on the NCT dataset.

Strong augmentation Following previous work (Grill et al., 2020; Chen et al., 2021c; Yang et al., 2022), we use similar strong data augmentation which contains random resized cropping, horizontal flipping, horizontal flipping, color jittering (Wu et al., 2018) with (brightness=0.4,

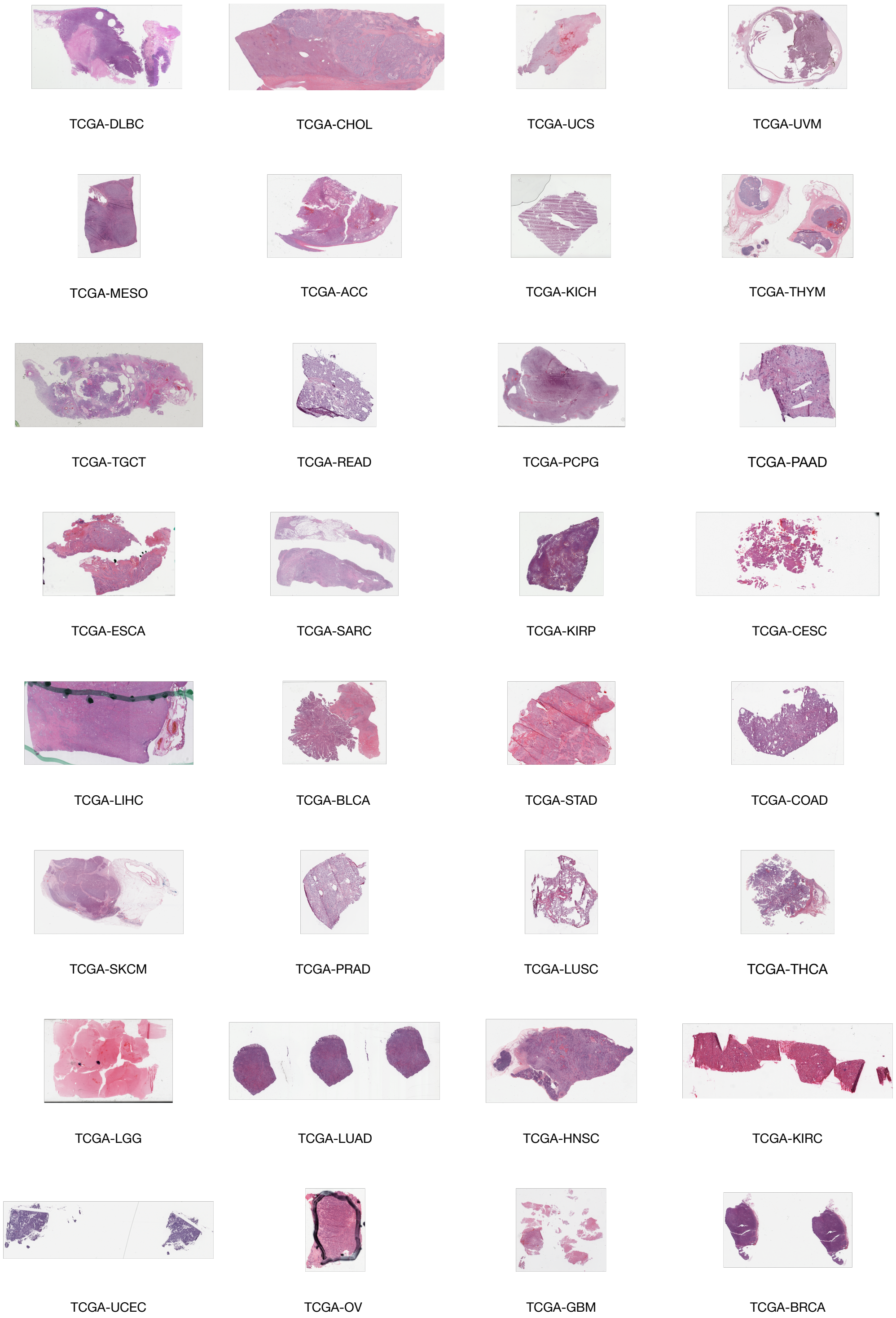


Figure 9: Randomly selected examples from TCGA dataset

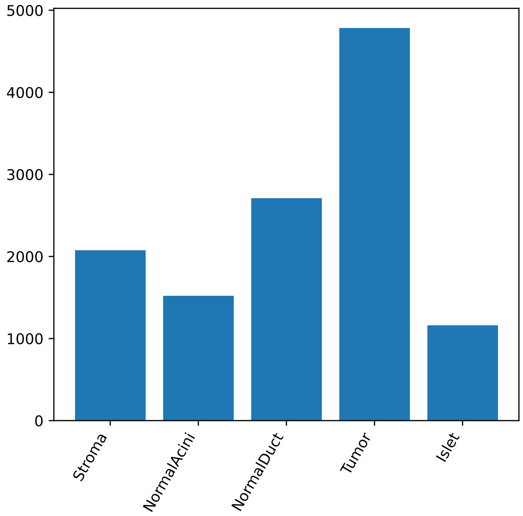


Figure 10: Patch number for each tissue for PDAC

contrast=0.4,saturation=0.4,hue=0.4,probability=0.8), grayscale conversion (Wu et al., 2018) with (probability=0.2), Gaussian blurring (Chen et al., 2020) with (kernel size=5, min=0.1, max=2.0, probability=0.5), and polarization (Grill et al., 2020) with (threshold=128, probability=0.2).

E Latent Augmentation

Latent augmentation (LA) was originally proposed in Yang et al. (2022) to improve the performance of the few-shot learning system in a simple unsupervised way. The pretrained feature extractor can only transfer parts of available knowledge in the pretraining datasets by the learned weights of the feature extractor. More transferable knowledge is inherent in the pretraining data representations.

In order to fully exploit the pretraining data, possible semantic shifts of clustered representations of the pretraining dataset are transferred to downstream tasks besides the pretrained feature extractor weights. The k-means clustering method is performed on the representations of pretraining datasets, which are generated by the pretrained feature extractor $\hat{e}$. Assume we obtain C clusters after clustering. The base dictionary $\mathcal{B} = \{(c_i, \Sigma_i)\}_{i=1}^{C}$ is constructed, where c_i is the i-th cluster prototype, i.e., mean representation of all samples in the cluster and Σ_i is the covariance matrix of the cluster. During downstream task testing, LA uses the original representation z to select the closest prototype from $\mathcal{B}$. We can get additive augmentation $\tilde{z} = z + \delta$, where δ is sampled from $\mathcal{N}(0, \Sigma_{i^*})$ and i^* is the index of closest prototype of z. The classifier of the downstream tasks is then trained on both the original representations and the augmented representations.

F Ablation Study

F.1 Set-input Models

Pooling architectures have been used in various set-input problems, e.g, 3D shape recognition (Shi et al., 2015; Su et al., 2015), learning the statistics of a set (Edwards & Storkey, 2016). Vinyals et al. (2015); Ilse et al. (2018) pool elements in a set by a weighted average with weights computed by the attention module. (Zaheer et al., 2017; Edwards & Storkey, 2016) proposed to aggregate embeddings of instances, extracted using a neural network, with pooling operations (e.g., mean, sum, max). This simple method satisfies the permutation invariant property and can work with any set size. Santoro et al. (2017) used a relational network to model all pairwise interactions of elements in a given set. Lee et al. (2019a) proposed to use the Transformer (Vaswani et al., 2017) to explicitly model higher-order interactions among the instances in a set.

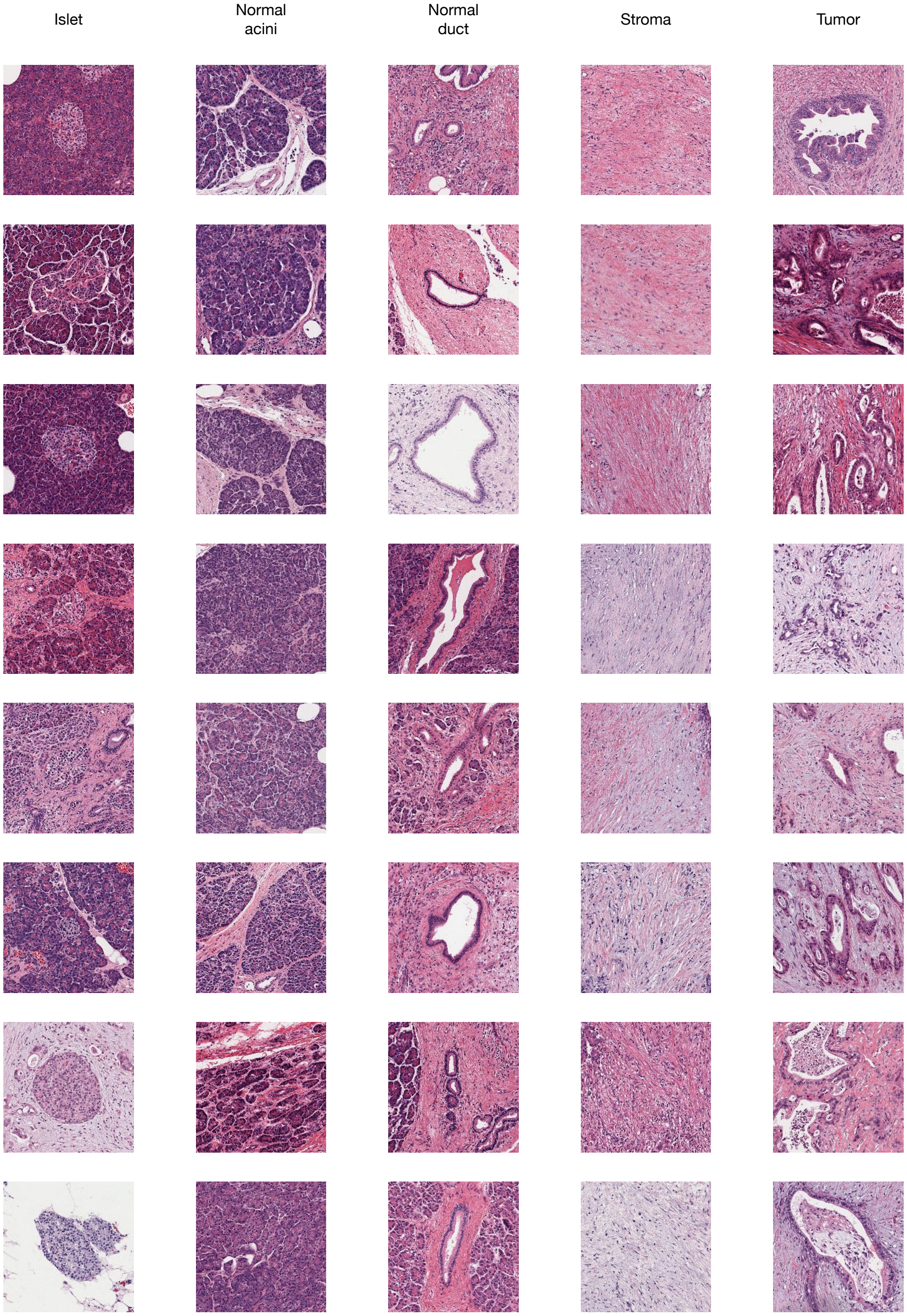


Figure 11: Randomly selected examples from each class of PDAC dataset.

We evaluate three set-input models for the FACILE-FSP model: attention-based MIL pooling (Ilse et al., 2018), Deep Set (Zaheer et al., 2017), and Set Transformer (Lee et al., 2019a). Attention-based MIL pooling uses a weighted average of instance embeddings from a set where weights are determined by a neural network. The attention-based MIL pooling corresponds to a version of attention (Lin et al., 2017; Raffel & Ellis, 2015). It has been adapted by Zhang et al. (2020b;a); Pal et al. (2021) in the context of H&E images. It uses a single fully connected layer and softmax with batch normalization and ReLU activation to predict the attention weights for instances. In the Deep Set model, each instance in a set is independently fed into a neural network that takes fixed-sized inputs. The extracted features are then aggregated using a pooling operation (i.e., mean, sum, or max). The final output is obtained by further non-linear operations. The simple architecture satisfies the permutation invariant property and can work with any set size. Set Transformer adapted the Transformer model for set data. It leverages the attention mechanism (Vaswani et al., 2017) to capture interactions between instances of the input set. It applies the idea of inducing points from the sparse Gaussian process literature to reduce quadratic complexity to linear in the size of the input set.

We train FACILE-FSP with three set-input models. The set size a is set to 5. In the attention-based MIL pooling model, we implemented the simple version, and use the single fc layer with softmax to predict attention weights from ResNet18 extracted features. For Deep Set model, we use two fc layers with ReLU activation functions in between to extract instance features before set pooling. In the Set Transformer, we use 4 attention heads and 3 inducing points.

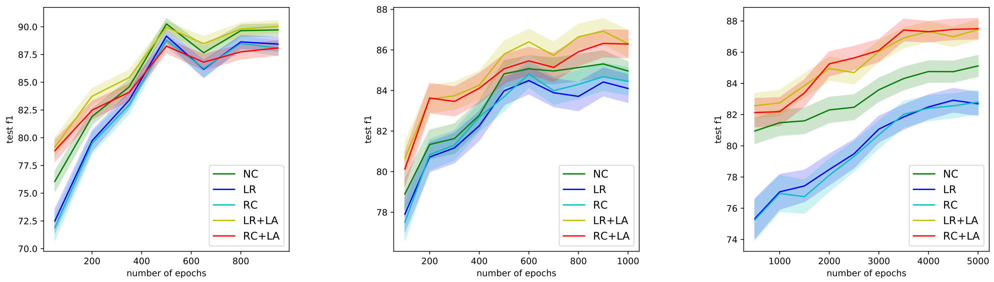
set-input model	NC	LR	RC	LR+LA	RC+LA
1-shot 5-way test on LC dataset					
Attention-based MIL pooling	70.53 ± 1.32	69.86 ± 1.39	69.75 ± 1.37	71.15 ± 1.31	70.31 ± 1.34
Deep Set	$\mathbf{77.84 \pm 1.16}$	$\mathbf{77.56 \pm 1.16}$	$\mathbf{77.56 \pm 1.17}$	$\mathbf{79.16 \pm 1.09}$	$\mathbf{77.38 \pm 1.18}$
Set Transformer	75.09 ± 1.30	73.57 ± 1.29	73.16 ± 1.33	74.03 ± 1.28	72.88 ± 1.34
5-shot 5-way test on LC dataset					
Attention-based MIL pooling	88.12 ± 0.59	81.60 ± 1.04	82.51 ± 0.97	89.18 ± 0.57	88.15 ± 0.65
Deep Set	90.35 ± 0.50	$\mathbf{90.91 \pm 0.47}$	$\mathbf{91.54 \pm 0.46}$	$\mathbf{91.68 \pm 0.50}$	$\mathbf{90.97 \pm 0.54}$
Set Transformer	$\mathbf{90.67 \pm 0.54}$	89.18 ± 0.61	89.02 ± 0.63	90.03 ± 0.59	88.71 ± 0.67
1-shot 3-way test on PAIP dataset					
Attention-based MIL pooling	50.98 ± 1.37	51.93 ± 1.35	51.91 ± 1.36	51.98 ± 1.36	52.39 ± 1.35
Deep Set	$\mathbf{52.04 \pm 1.25}$	$\mathbf{53.27 \pm 1.25}$	$\mathbf{54.19 \pm 1.26}$	$\mathbf{52.66 \pm 1.25}$	$\mathbf{52.79 \pm 1.23}$
Set Transformer	48.81 ± 1.21	50.08 ± 1.24	50.75 ± 1.23	50.03 ± 1.23	49.41 ± 1.20
5-shot 3-way test on PAIP dataset					
Attention-based MIL pooling	67.04 ± 1.00	66.06 ± 1.17	66.61 ± 1.10	$\mathbf{70.19 \pm 0.87}$	$\mathbf{70.54 \pm 0.81}$
Deep Set	$\mathbf{69.42 \pm 0.85}$	$\mathbf{69.93 \pm 0.92}$	$\mathbf{70.52 \pm 0.87}$	69.96 ± 0.84	68.39 ± 0.84
Set Transformer	66.61 ± 0.91	67.57 ± 0.95	67.78 ± 0.95	68.24 ± 0.85	67.20 ± 0.86
1-shot 9-way test on NCT dataset					
Attention-based MIL pooling	60.04 ± 1.40	64.53 ± 1.29	64.81 ± 1.31	64.00 ± 1.34	66.66 ± 1.32
Deep Set	$\mathbf{68.21 \pm 1.30}$	68.17 ± 1.31	$\mathbf{68.69 \pm 1.30}$	$\mathbf{69.24 \pm 1.28}$	$\mathbf{68.18 \pm 1.33}$
Set Transformer	67.76 ± 1.31	$\mathbf{68.52 \pm 1.30}$	68.55 ± 1.28	68.33 ± 1.28	67.72 ± 1.28
5-shot 9-way test on NCT dataset					
Attention-based MIL pooling	81.94 ± 0.75	82.40 ± 0.72	84.46 ± 0.65	86.49 ± 0.62	$\mathbf{87.66 \pm 0.59}$
Deep Set	85.18 ± 0.60	85.87 ± 0.60	87.11 ± 0.56	87.06 ± 0.61	85.81 ± 0.66
Set Transformer	$\mathbf{86.45 \pm 0.62}$	$\mathbf{87.74 \pm 0.59}$	$\mathbf{87.97 \pm 0.58}$	$\mathbf{88.00 \pm 0.59}$	86.92 ± 0.61

Table 11: Performance of FACIEL-FSP-WSI with three different set-input models; average F1 and CI are reported.

From table 11, we conclude that none of the 3 set-input models used in FACILE-FSP is consistently better than the other set-input models. The Deep Set model achieves the highest average F1 score with more tasks.

F.2 Learning Curve

To validate the adequacy of training for all models, we assess the intermediate checkpoints of each pretraining model on the LC dataset. The learning curves and confidence intervals (CI) of FACILE-FSP, FSP-Patch, and SimSiam are displayed in figure 12. Upon careful examination of the learning curves in figure 12, we observe conclusive evidence of complete training for all models, as they have reached convergence.



(a) FACILE-FSP model

(b) FSP-Patch model

(c) SimSiam model

Figure 12: Learning curves of FACILE-FSP model, FSP-Patch model, and SimSiam. The mean F1 score and CI of 5 few-shot models tested on the LC dataset with 5-shot are shown with curves.

F.3 Input Set Size

To examine the impact of input set size on downstream tasks, we conduct pretraining experiments using FACILE-FSP on the TCGA dataset with varying input set sizes. The resulting feature map e from the trained FACILE-FSP is then evaluated on LC, PAIP, and NCT datasets with shot numbers 1 and 5. The corresponding performances are reported in table 12.

Observing table 12, we find that models with an input set size of 5 consistently demonstrate superior performance for LC and PAIP datasets. While slight improvements are observed for larger input set sizes, they are not substantial. Conversely, for the NCT dataset, as presented in table 12, the best performance is attained when the input set size is 10.

G Contrastive and Non-contrastive Learning Models

Self-supervised learning achieves promising results on multiple visual tasks (Bachman et al., 2019; He et al., 2020; Chen et al., 2020; Grill et al., 2020; Caron et al., 2020; Chen & He, 2021). Contrastive learning method avoid collapse by encouraging the representations to be far apart for views from different images. Henaff (2020); He et al. (2020); Misra & Maaten (2020); Chen et al. (2020) implemented instance discrimination, in which a pair of augmented views from the same image are positive and others are negative. Caron et al. (2020; 2018) contrasted different cluster of positives. Non-contrastive models (Grill et al., 2020; Richemond et al., 2020; Chen & He, 2021) removed the reliance on negatives. These non-contrastive models achieved strong results in the ImageNet (Deng et al., 2009) pretraining setting. SimSiam (Chen & He, 2021) works with typical batches and does not rely on large-batch training, which makes it preferable for academics and practitioners with low computation resources.

In this section, some contrastive learning and non-contrastive learning models, e.g., SimCLR, SupCon, and SimSiam, that are used in this paper are explained. Details of implementation are provided. There are three main components in SimCLR and SupCon framework. We follow the notation of Khosla et al. (2020) in this section to explain SimCLR and SupCon.

- Data augmentation $Aug(\cdot)$. For each input sample x, the augmentation module generates two random augmented views, i.e., $\tilde{x} \sim Aug(x)$. The augmentation schedules used in this paper are explained in section D.
- Encoder $Enc(\cdot)$. The encoder extracts a representation vector $r = Enc(\tilde{x})$. The pair of augmented views are separately fed to the same encoder and generate a pair of representations. The r is normalized to the unit hypersphere.
- Projection head $Proj(\cdot)$. It maps r to a vector $z = Proj(r)$. We instantiate $Proj(\cdot)$ as a multi-layer perceptron (MLP) with a single hidden layer of size 512 and output vector size of 512. We also normalize the output to the unit hypersphere.

set size	2	5	10	15
1-shot 5-way test on LC dataset				
NC	75.29 ± 1.33	$\mathbf{77.84 \pm 1.16}$	74.88 ± 1.36	75.25 ± 1.29
LR	73.72 ± 1.33	$\mathbf{77.56 \pm 1.16}$	73.84 ± 1.29	74.00 ± 1.27
RC	74.10 ± 1.34	$\mathbf{77.56 \pm 1.17}$	73.42 ± 1.31	73.42 ± 1.29
LR+LA	75.27 ± 1.28	$\mathbf{79.16 \pm 1.09}$	74.41 ± 1.31	74.92 ± 1.26
RC+LA	74.36 ± 1.33	$\mathbf{77.38 \pm 1.18}$	72.60 ± 1.34	73.16 ± 1.32
5-shot 5-way test on LC dataset				
NC	90.62 ± 0.56	90.35 ± 0.50	90.62 ± 0.57	$\mathbf{90.83 \pm 0.55}$
LR	89.41 ± 0.63	$\mathbf{90.91 \pm 0.47}$	89.80 ± 0.59	89.63 ± 0.60
RC	89.11 ± 0.63	$\mathbf{91.54 \pm 0.46}$	89.26 ± 0.61	89.25 ± 0.60
LR+LA	90.46 ± 0.58	$\mathbf{91.68 \pm 0.50}$	90.29 ± 0.57	90.46 ± 0.56
RC+LA	89.64 ± 0.63	$\mathbf{90.97 \pm 0.54}$	88.52 ± 0.66	89.00 ± 0.64
NC	48.95 ± 1.24	52.04 ± 1.25	51.72 ± 1.22	$\mathbf{52.46 \pm 1.20}$
LR	50.55 ± 1.22	53.27 ± 1.25	52.33 ± 1.25	$\mathbf{53.38 \pm 1.23}$
RC	50.14 ± 1.25	$\mathbf{54.19 \pm 1.26}$	53.04 ± 1.24	52.68 ± 1.25
LR+LA	50.12 ± 1.22	52.66 ± 1.25	52.96 ± 1.21	$\mathbf{53.41 \pm 1.21}$
RC+LA	49.91 ± 1.22	$\mathbf{52.79 \pm 1.23}$	51.67 ± 1.17	51.51 ± 1.20
5-shot 3-way test on PAIP dataset				
NC	66.99 ± 0.93	$\mathbf{69.42 \pm 0.85}$	69.10 ± 0.91	69.08 ± 0.87
LR	68.11 ± 0.94	69.93 ± 0.92	$\mathbf{70.30 \pm 0.90}$	69.28 ± 0.90
RC	68.63 ± 0.91	$\mathbf{70.52 \pm 0.87}$	70.45 ± 0.87	70.12 ± 0.90
LR+LA	69.03 ± 0.83	69.96 ± 0.84	$\mathbf{70.25 \pm 0.81}$	70.00 ± 0.81
RC+LA	67.32 ± 0.83	$\mathbf{68.39 \pm 0.84}$	68.35 ± 0.83	67.70 ± 0.81
NC	66.31 ± 1.36	68.21 ± 1.30	$\mathbf{72.44 \pm 1.25}$	72.05 ± 1.27
LR	68.55 ± 1.32	68.17 ± 1.31	$\mathbf{72.62 \pm 1.25}$	72.14 ± 1.27
RC	68.58 ± 1.32	68.69 ± 1.30	$\mathbf{72.60 \pm 1.25}$	72.04 ± 1.27
LR+LA	67.42 ± 1.33	69.24 ± 1.28	$\mathbf{72.18 \pm 1.26}$	71.92 ± 1.27
RC+LA	65.87 ± 1.36	68.18 ± 1.33	$\mathbf{69.98 \pm 1.31}$	69.88 ± 1.28
5-shot 9-way test on NCT dataset				
NC	85.28 ± 0.72	85.18 ± 0.60	$\mathbf{88.25 \pm 0.56}$	88.22 ± 0.57
LR	86.39 ± 0.69	85.87 ± 0.60	$\mathbf{88.80 \pm 0.55}$	88.55 ± 0.55
RC	87.03 ± 0.66	87.11 ± 0.56	$\mathbf{89.25 \pm 0.52}$	89.02 ± 0.54
LR+LA	86.85 ± 0.65	87.06 ± 0.61	88.52 ± 0.59	$\mathbf{88.93 \pm 0.55}$
RC+LA	85.60 ± 0.70	85.81 ± 0.66	87.40 ± 0.63	$\mathbf{87.74 \pm 0.59}$

Table 12: Abation on set size; models tested on LC, PAIP, and NCT dataset; average F1 and CI are reported.

For a set of N randomly sampled sample/label pairs, $\{(x_k, y_k)\}_{k=1}^{N}$. The corresponding batch used for training consists of $2N$ pairs, $\{(\tilde{x}_l, \tilde{y}_l)\}_{l=1}^{2N}$, where $\tilde{x}_{2k-1}$ and $\tilde{x}_{2k}$ are two random augmented views of x_k and $\tilde{y}_{2k-1} = \tilde{y}_{2k} = y_k$.

G.1 SimCLR

Let $i \in I \equiv \{1 \ldots 2N\}$ be the index of an arbitrary augmented sample and let $j(i)$ be the index of the other augmented sample originating from the same source sample. The abstraction of SimCLR structure can be found in figure 13. In SimCLR, the loss takes the following form.

$$\mathcal{L}^{\text{self}} = \sum_{i \in I} \mathcal{L}_i^{\text{self}} = -\sum_{i \in I} \log \frac{\exp\left(z_i \cdot z_{j(i)}/\tau\right)}{\sum_{a \in A(i)} \exp\left(z_i \cdot z_a/\tau\right)} \tag{2}$$

where τ is the temperature parameter. $A(i) \equiv I \backslash \{i\}$. The denominator has a total of $2N - 1$ terms.

In this paper, the τ is always set to 0.07. The patches are augmented randomly by the augmentation module described in section D. We use an MLP as a projection head with two fully-connected (fc) layers, a hidden dimension of 512, and an output dimension of 512.

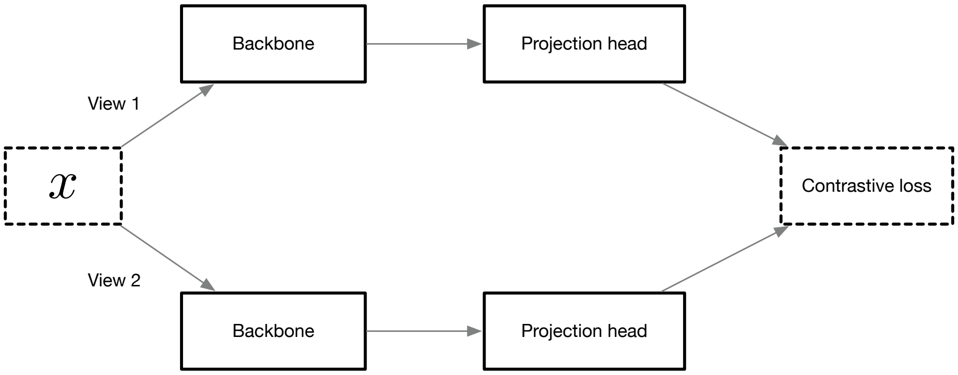


Figure 13: Abstraction of SimCLR structure

G.2 SupCon

For supervised learning, the contrastive loss in equation 2 cannot handle class discrimination (Khosla et al., 2020). Khosla et al. (2020) proposed two straightforward ways, as shown in equation 3 and equation 4, to generalize equation 2 to incorporate supervison.

$$\mathcal{L}_{\text{out}}^{\text{sup}} = \sum_{i \in I} \mathcal{L}_{\text{out},i}^{\text{sup}} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp\left(z_i \cdot z_p/\tau\right)}{\sum_{a \in A(i)} \exp\left(z_i \cdot z_a/\tau\right)} \tag{3}$$

$$\mathcal{L}_{in}^{\text{sup}} = \sum_{i \in I} \mathcal{L}_{in,i}^{\text{sup}} = \sum_{i \in I} -\log \left\{ \frac{1}{|P(i)|} \sum_{p \in P(i)} \frac{\exp\left(z_i \cdot z_p/\tau\right)}{\sum_{a \in A(i)} \exp\left(z_i \cdot z_a/\tau\right)} \right\} \tag{4}$$

Here $P(i) \equiv \{p \in A(i) : \tilde{y}_p = \tilde{y}_i\}$ is the set of indices of all positives in the batch distinct from i. The authors showed that $\mathcal{L}_{\text{in}}^{\text{sup}} \leq \mathcal{L}_{\text{out}}^{\text{sup}}$ and $\mathcal{L}_{\text{out}}^{\text{sup}}$ is the superior supervised loss function. Thus, we use SupCon with equation 3 as the default loss. The τ is also set to 0.07.

In our model FACILE-SupCon-WSI, the input sample is a set of randomly sampled patches and labels are slide properties, i.e., organs or TCGA projects. Each patch is augmented randomly by the augmentation module described in section D. The feature map e and set function g work as the encoder $Enc(\cdot)$. We also use an MLP as a projection head with two fc layers, a hidden dimension of 512, and an output dimension of 512.

G.3 SimSiam

Non-contrastive models, e.g., SimSiam and BYOL, achieve strong results in typical ImageNet (Deng et al., 2009) pretraining setting (Chen & He, 2021; Grill et al., 2020; Li et al., 2022). Among the non-contrastive models, SimSiam removes the negatives and uses stop-grad to avoid collapse. Besides, it trains faster, requires less GPU memory, and works well with small batch size (Chen & He, 2021; Li et al., 2022), which makes it extremely appealing to academics.

The abstraction of SimSiam structure is shown in figure 14. Given two augmented views $\tilde{x}_1$ and $\tilde{x}_2$ of the same image x, SimSiam learns to use $\tilde{x}_1$ to predict the representation of $\tilde{x}_2$. Specifically, $\tilde{x}_1$ is passed into the online backbone network on the upper. The $\tilde{x}_2$ is passed into the target backbone network on the lower. The outputs of the two backbone networks are passed to the projection MLPs and then a prediction MLP is used to predict the projected representation of $\tilde{x}_2$ from the projected representation of $\tilde{x}_1$. SimSiam uses the same network for the online and target backbone and projection networks.

In our paper, the projection MLP has 3 fc layers with a hidden dimension of 512 and an output dimension of 512. It has batch normalization (BN) applied to each fc including its output fc. The prediction MLP also has BN applied to its hidden fc. Its output fc does not have BN or ReLU. The prediction MLP has 2 layers.

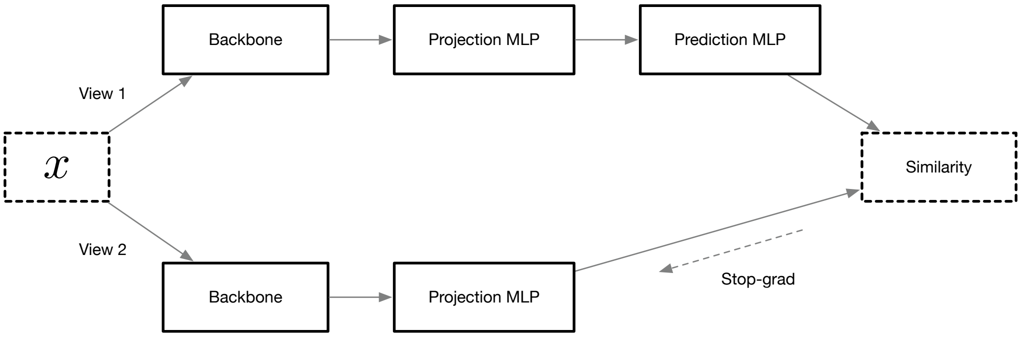


Figure 14: Abstraction of SimSiam structure

H Excess Risk Bound of FACILE

Our proof framework follows closely the work of Robinson et al. (2020). We consider the setting where we have some coarse-grained labels of some sets, rather than instances and the downstream classifiers only use the learned embeddings to train and test on the downstream tasks. /Robinson et al. (2020) considers a different setting where each instance has a weak label and a strong label, and the strong label predictor learns to predict the strong labels from the instances and their corresponding embeddings learned with weak labels. The diagram of only using trained embeddings for downstream tasks is more often used in self-supervised learning and representation learning for FSL literature (Du et al., 2020; Yang et al., 2021; Bachman et al., 2019; He et al., 2020; Chen et al., 2020; Grill et al., 2020; Caron et al., 2020; Chen & He, 2021). The coarse-grained data contains useful information, which is characterized by our defined Lipschitzness, to pretrain a instance feature map that can be leveraged for downstream FSL. We include the full proof of our key result as follows.

In order to prove theorem 4, we first split the excess risk by the following proposition.

Proposition 5. *Suppose that f^* is L-Lipschitz relative to $\mathcal{E}$. The excess risk $\mathbb{E}\left[\ell^{\text{fg}}_{\hat{f}\circ\hat{e}}(X,Y) - \ell^{\text{fg}}_{f^*\circ e^*}(X,Y)\right]$ is bounded by,*

$$2L\text{Rate}_m(\ell^{\text{cg}}, P_{S,W}, \mathcal{E}) + \text{Rate}_n(\ell^{\text{fg}}, \hat{P}_{Z,Y}, \mathcal{F})$$

Proof. We split the excess risk into three parts

$$\begin{aligned}
&\mathbb{E}_{P_{X,Y}}\left[\ell^{\text{fg}}_{\hat{f}\circ\hat{e}}(X,Y) - \ell^{\text{fg}}_{f^*\circ e^*}(X,Y)\right] \\
=&\mathbb{E}_{P_{X,Y}}\left[\ell^{\text{fg}}_{\hat{f}\circ\hat{e}}(X,Y) - \ell^{\text{fg}}_{f^*\circ\hat{e}}(X,Y)\right] + \mathbb{E}_{P_{X,Y}}\left[\ell^{\text{fg}}_{f^*\circ\hat{e}}(X,Y) - \ell^{\text{fg}}_{f^*\circ e_0}(X,Y)\right] \\
&\quad + \mathbb{E}_{P_{X,Y}}\left[\ell^{\text{fg}}_{f^*\circ e_0}(X,Y) - \ell^{\text{fg}}_{f^*\circ e^*}(X,Y)\right]
\end{aligned}$$

For the second term and third term, relative Lipschitzness of f^* to $\mathcal{E}$ delivers

$$\begin{aligned}
\mathbb{E}_{P_{X,Y}}\left[\ell^{\text{fg}}_{f^*\circ\hat{e}}(X,Y) - \ell^{\text{fg}}_{f^*\circ e_0}(X,Y)\right] &= \mathbb{E}_{P_{X,Y,S,W}}\left[\ell^{\text{fg}}_{f^*\circ\hat{e}}(X,Y) - \ell^{\text{fg}}_{f^*\circ e_0}(X,Y)\right] \\
&\leq L\mathbb{E}_{P_{X,Y,S,W}}\ell^{\text{cg}}\left(g_{\hat{e}}\circ\hat{e}(S), g_{e_0}\circ e_0(S)\right) \\
&= L\mathbb{E}_{P_{S,W}}\ell^{\text{cg}}\left(g_{\hat{e}}\circ\hat{e}(S), g_{e_0}\circ e_0(S)\right),
\end{aligned}$$

$$\begin{aligned}
\mathbb{E}_{P_{X,Y}}\left[\ell^{\text{fg}}_{f^*\circ e_0}(X,Y) - \ell^{\text{fg}}_{f^*\circ e^*}(X,Y)\right] &= \mathbb{E}_{P_{X,Y,S,W}}\left[\ell^{\text{fg}}_{f^*\circ e_0}(X,Y) - \ell^{\text{fg}}_{f^*\circ e^*}(X,Y)\right] \\
&\leq L\mathbb{E}_{P_{X,Y,S,W}}\ell^{\text{cg}}\left(g_{e_0}\circ e_0(S), g_{e^*}\circ e^*(S)\right) \\
&= L\mathbb{E}_{P_{S,W}}\ell^{\text{cg}}\left(g_{e_0}\circ e_0(S), g_{e^*}\circ e^*(S)\right)
\end{aligned}$$

Since e^* attains minimal risk and $W = g_{e_0}\circ e_0(S)$, the sum of the two terms can be bounded by,

$$\begin{aligned}&L\mathbb{E}_{P_{S,W}}\ell^{\mathrm{cg}}\left(g_{\hat{e}}\circ\hat{e}(S),g_{e_0}\circ e_0(S)\right)+L\mathbb{E}_{P_{S,W}}\ell^{\mathrm{cg}}\left(g_{e_0}\circ e_0(S),g_{e^*}\circ e^*(S)\right)\\ \leq &2L\mathbb{E}_{P_{S,W}}\ell^{\mathrm{cg}}\left(g_{\hat{e}}\circ\hat{e}(S),W\right)\leq 2L\mathrm{Rate}_m(\ell^{\mathrm{cg}},P_{S,W},\mathcal{E})\end{aligned}$$

By combining the bounds on the three terms we can get the claim.

The central condition is well-known to yield fast rates for supervised learning (Van Erven et al., 2015). It directly implies that we could learn a map $Z\rightarrow Y$ with $\widetilde{\mathcal{O}}(1/n)$ excess risk. The difficulty is that at test time we would need access to latent value $Z=e(X)$. To circumnavigate this hurdle, we replace e_0 with $\hat{e}$ and solve the supervised learning problem $(\ell^{\mathrm{fg}},\hat{P}_{Z,Y},\mathcal{F})$.

It is not clear whether this surrogate problem satisfies the central condition. We show that $(\ell^{\mathrm{fg}},\hat{P}_{Z,Y},\mathcal{F})$ indeed satisfies a weak central condition and shows weak central condition still enables strong excess risk guarantees.

Following Robinson et al. (2020); Van Erven et al. (2015), we define the central condition on $\mathcal{F}$.

Definition 6. (The central condition). A learning problem $(\ell^{\mathrm{fg}},P_{Z,Y},\mathcal{F})$ on $\mathcal{Z}\times\mathcal{Y}$ is said to satisfy the ϵ-weak η-central condition if there exists an $f^*\in\mathcal{F}$ such that

$$\mathbb{E}_{(Z,Y)\sim P_{Z,Y}}\left[e^{\eta(\ell^{\mathrm{fg}}_{f^*}(Z,Y)-\ell^{\mathrm{fg}}_{f}(Z,Y))}\right]\leq e^{\eta\epsilon}$$

for all $f\in\mathcal{F}$. The 0-weak central condition is known as the strong central condition.

Capturing relatedness of pretraining and downstream task with the central condition. Intuitively, the strong central condition requires that the minimal risk model f^* attains a higher loss than $f\in\mathcal{F}$ on a set of Z,Y with an exponentially small probability. This is likely to happen when Z is highly predictive of Y so that the probability of $P(Y|Z)$ concentrates in a single location for most Z. If f^* in $\mathcal{F}$ such that $f^*(Z)$ maps into this concentration, $\ell^{\mathrm{fg}}_{f^*}(Z,Y)$ will be close to zero most of the time.

We assume that the strong central condition holds for the learning problem $(\ell^{\mathrm{fg}},P_{Z,Y},\mathcal{F})$ where $Z=e_0(X)$. Similar to Robinson et al. (2020), we split the learning procedure into two supervised tasks as depicted in Algorithm 1. In the algorithm, we replace $(\ell^{\mathrm{fg}},P_{Z,Y},\mathcal{F})$ with $(\ell^{\mathrm{fg}},\hat{P}_{Z,Y},\mathcal{F})$.

We will show that $(\ell^{\mathrm{fg}},\hat{P}_{Z,Y},\mathcal{F})$ satisfies the weak central condition.

Proposition 7. *Assume that $\ell^{\mathrm{cg}}(w,w')=\mathbb{1}\{w\neq w'\}$ and that ℓ^{fg} is bounded by $B>0$, $\mathcal{F}$ is L-Lipschitz relative to $\mathcal{E}$, and that $(\ell^{\mathrm{fg}},P_{Z,Y},\mathcal{F})$ satisfies ϵ-weak central condition. Then $(\ell^{\mathrm{fg}},\hat{P}_{Z,Y},\mathcal{F})$ satisfies the $\epsilon+\mathcal{O}\left(\frac{\exp(\eta B)}{\eta}\mathrm{Rate}_m\left(\mathcal{E},P_{S,W}\right)\right)$-weak central condition with probability at least $1-\delta$.*

Proof. Note that

$$\frac{1}{\eta}\log\mathbb{E}_{\hat{P}_{Z,Y}}\exp\left(\eta(\ell^{\mathrm{fg}}_{f^*}-\ell^{\mathrm{fg}}_{f})\right)=\frac{1}{\eta}\log\mathbb{E}_{P_{X,Y}}\exp\left(\eta(\ell^{\mathrm{fg}}_{f^*\circ\hat{e}}-\ell^{\mathrm{fg}}_{f\circ\hat{e}})\right)$$

To prove that $(\ell^{\mathrm{fg}},\hat{P}_{Z,Y},\mathcal{F})$ satisfies the central condition we therefore need to bound $\frac{1}{\eta}\log\mathbb{E}_{P_{X,Y}}\exp\left(\eta(\ell^{\mathrm{fg}}_{f^*\circ\hat{e}}-\ell^{\mathrm{fg}}_{f\circ\hat{e}})\right)$ by some constant.

$$
\begin{aligned}
&\frac{1}{\eta}\log \mathbb{E}_{P_{X,Y}} \exp\Big(\eta(\ell^{\text{fg}}_{f^*\circ\hat{e}} - \ell^{\text{fg}}_{f\circ\hat{e}})\Big)\\
=&\frac{1}{\eta}\log \mathbb{E}_{P_{X,Y,S,W}} \exp\Big(\eta(\ell^{\text{fg}}_{f^*\circ\hat{e}} - \ell^{\text{fg}}_{f\circ\hat{e}})\Big)\\
=&\frac{1}{\eta}\log \mathbb{E}_{P_{X,Y,S,W}} \Big[\exp(\eta(\ell^{\text{fg}}_{f^*\circ\hat{e}} - \ell^{\text{fg}}_{f\circ\hat{e}}))\mathbb{1}\{\hat{g}_{\hat{e}}\circ\hat{e}(S)=W\}\Big]+\\
&\qquad\frac{1}{\eta}\log \mathbb{E}_{P_{X,Y,S,W}} \Big[\exp(\eta(\ell^{\text{fg}}_{f^*\circ\hat{e}} - \ell^{\text{fg}}_{f\circ\hat{e}}))\mathbb{1}\{\hat{g}_{\hat{e}}\circ\hat{e}(S)\neq W\}\Big]\\
=&\underbrace{\frac{1}{\eta}\log \mathbb{E}_{P_{X,Y,S,W}} \Big[\exp(\eta(\ell^{\text{fg}}_{f^*\circ e_0} - \ell^{\text{fg}}_{f\circ e_0}))\mathbb{1}\{\hat{g}_{\hat{e}}\circ\hat{e}(S)=W\}\Big]}_{\text{first term}}+\\
&\qquad\underbrace{\frac{1}{\eta}\log \mathbb{E}_{P_{X,Y,S,W}} \Big[\exp(\eta(\ell^{\text{fg}}_{f^*\circ\hat{e}} - \ell^{\text{fg}}_{f\circ\hat{e}}))\mathbb{1}\{\hat{g}_{\hat{e}}\circ\hat{e}(S)\neq W\}\Big]}_{\text{second term}}
\end{aligned}
$$

The third line follows from the fact that for any f in the event $\{\hat{g}_{\hat{e}}\circ\hat{e}(S)=W\}$ we have $\ell^{\text{fg}}_{f\circ\hat{g}} = \ell^{\text{fg}}_{f\circ g_0}$.

This is because $|\ell^{\text{fg}}_{f\circ\hat{e}}(X,Y) - \ell^{\text{fg}}_{f\circ e_0}(X,Y)| \leq L\ell^{\text{cg}}(g_{\hat{e}}\circ\hat{e}(S), g_{e_0}\circ e_0(S)) = L\ell^{\text{cg}}(W,W)=0$.

We get $\frac{1}{\eta}\log \mathbb{E}_{P_{X,Y,S,W}}\Big[\exp(\eta(\ell^{\text{fg}}_{f^*\circ e_0} - \ell^{\text{fg}}_{f\circ e_0}))\Big]$ after we drop the $\mathbb{1}\{\hat{g}_{\hat{e}}\circ\hat{e}(S)=W\}$. It is bounded by ϵ with the weak central condition. The second term is bounded by

$$
\begin{aligned}
&\frac{1}{\eta}\log \mathbb{E}_{P_{X,Y,S,W}} \Big[\exp(\eta(\ell^{\text{fg}}_{f^*\circ\hat{e}} - \ell^{\text{fg}}_{f\circ\hat{e}}))\mathbb{1}\{\hat{g}_{\hat{e}}\circ\hat{e}(S)\neq W\}\Big]\\
\leq&\frac{1}{\eta}\log \mathbb{E}_{P_{X,Y,S,W}} [\exp(\eta B)\mathbb{1}\{\hat{g}_{\hat{e}}\circ\hat{e}(S)\neq W\}]\\
=&\frac{1}{\eta}\log \mathbb{E}_{P_{S,W}} [\exp(\eta B)\mathbb{1}\{\hat{g}_{\hat{e}}\circ\hat{e}(S)\neq W\}]\\
<&\frac{1}{\eta}\mathbb{E}_{P_{S,W}} [\exp(\eta B)\mathbb{1}\{\hat{g}_{\hat{e}}\circ\hat{e}(S)\neq W\}]\\
=&\frac{\exp(\eta B)}{\eta}P_{S,W}(\hat{g}_{\hat{e}}\circ\hat{e}(S)\neq W)\\
=&\frac{\exp(\eta B)}{\eta}\text{Rate}_m(\ell^{\text{cg}}, P_{S,W}, \mathcal{E})
\end{aligned}
$$

The first inequality uses the fact that ℓ^{fg} is bounded by B. The forth line is because that $\log x < x$. By combining this bound with ϵ bound on the first term we can get the claimed result of proposition 7.

The proof of the main theorem further relies on a proposition provided by Robinson et al. (2020), as we show below:

Proposition 8. *Robinson et al. (2020) Suppose $(\ell^{\text{fg}}, Q_{Z,Y}, \mathcal{F})$ satisfies the ϵ-weak central condition, ℓ^{fg} is bounded by $B>0$, $\mathcal{F}$ is L'-Lipschitz in its d-dimensional parameters in the l_2 norm, $\mathcal{F}$ is contained in Euclidean ball of radius R, and $\mathcal{Y}$ is compact. Then when $\mathcal{A}_n(\ell^{\text{fg}}, Q_{Z,Y}, \mathcal{F})$ is ERM, the excess risk $\mathbb{E}_{Z,Y\sim Q_{Z,Y}}\Big[\ell^{\text{fg}}_{\hat{f}}(Z,Y) - \ell^{\text{fg}}_{f^*}(Z,Y)\Big]$ is bounded by,*

$$
\mathcal{O}\left(V\frac{d\log\frac{RL'}{\epsilon}+\log\frac{1}{\delta}}{n}+V\epsilon\right)
$$

with probability at least $1-\delta$, where $V=B+\epsilon$.

Proof of the main theorem: If $m = \Omega(n^{\beta})$, the $\mathrm{Rate}_m(\ell^{\mathrm{cg}}, P_{S,W}, \mathcal{E}) = \mathcal{O}(\frac{1}{m^{\alpha}}) = \mathcal{O}(\frac{1}{n^{\alpha\beta}})$. proposition 7 concludes that $(\ell^{\mathrm{fg}}, \hat{P}_{Z,Y}, \mathcal{F})$ satisfies the $\mathcal{O}(\frac{1}{n^{\alpha\beta}})$-weak central condition with probability at least $1-\delta$. Thus by proposition 8, we can get $\mathrm{Rate}_n(\ell^{\mathrm{fg}}, \hat{P}_{Z,Y}, \mathcal{F}) = \mathcal{O}\left(\frac{d\alpha\beta \log RL'n + \log\frac{1}{\delta}}{n} + \frac{B}{n^{\alpha\beta}}\right)$. Combining bounds with proposition 5 we conclude that

$$\begin{aligned}\mathbb{E}\left[\ell^{\mathrm{fg}}_{\hat{f}\circ\hat{e}}(X,Y) - \ell^{\mathrm{fg}}_{f^*\circ e^*}(X,Y)\right] \leq & 2L\mathrm{Rate}_m(\ell^{\mathrm{cg}}, P_{S,W}, \mathcal{E}) + \mathrm{Rate}_n(\ell^{\mathrm{fg}}, \hat{P}_{Z,Y}, \mathcal{F}) \\ \leq & \mathcal{O}\left(\frac{d\alpha\beta \log RL'n + \log\frac{1}{\delta}}{n} + \frac{B}{n^{\alpha\beta}} + \frac{2L}{n^{\alpha\beta}}\right)\end{aligned}$$