

THE UNIVERSITY OF CHICAGO

MACHINE LEARNING FOR HISTOPATHOLOGY IMAGES IN LOW-DATA REGIMES

A DISSERTATION SUBMITTED TO  
THE FACULTY OF THE DIVISION OF THE PHYSICAL SCIENCES  
IN CANDIDACY FOR THE DEGREE OF  
DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE

BY  
RENYU ZHANG

CHICAGO, ILLINOIS

AUGUST 2024

Copyright © 2024 by Renyu Zhang  
All Rights Reserved

To my beloved wife, Na Chen, whose steadfast confidence in my abilities and unwavering support are the cornerstones of every achievement, continually motivating me to improve.

To my parents, Hengxiang Xu and Qingjie Zhang, whose unconditional love and tireless dedication have shaped my values and inspire me to strive for greatness.

# TABLE OF CONTENTS

LIST OF FIGURES . . . . .	viii
LIST OF TABLES . . . . .	xi
ACKNOWLEDGMENTS . . . . .	xii
ABSTRACT . . . . .	xv
1 INTRODUCTION . . . . .	1
1.1 Motivation . . . . .	3
1.2 Summary of Contributions . . . . .	4
1.3 Thesis Organization . . . . .	8
1.4 List of Publications . . . . .	9
2 BACKGROUND AND RELATED WORKS . . . . .	11
2.1 Histopathology Images . . . . .	11
2.2 Caption Prediction . . . . .	12
2.3 Active Learning . . . . .	13
2.4 Few-shot Learning . . . . .	15
2.5 Self-supervised Learning . . . . .	17
3 CAPTION GENERATION FOR HISTOPATHOLOGY IMAGES . . . . .	20
3.1 Motivation . . . . .	20
3.2 Methods . . . . .	22
3.2.1 Overview . . . . .	22
3.2.2 Metric Learning with Triplet Loss . . . . .	23
3.2.3 Neural Network Architecture . . . . .	25
3.2.4 Data Augmentation and Hyperparameter Settings . . . . .	27
3.2.5 Cohort . . . . .	27
3.3 Results on Real Data . . . . .	29
3.3.1 Results on Caption Generation . . . . .	29
3.3.2 Results on Metric Learning . . . . .	29
3.3.3 Results on Clustering . . . . .	30
3.3.4 Results on Visualization . . . . .	31
3.4 Discussion . . . . .	32
4 HYPERBOLIC ATTENTION MODEL FOR HISTOPATHOLOGY IMAGES . . . . .	35
4.1 Motivation . . . . .	35
4.2 Method . . . . .	37
4.2.1 Poincaré Ball Model . . . . .	38
4.2.2 Möbius Addition . . . . .	38
4.2.3 Exponential and Logarithmic Maps . . . . .	38

4.2.4	Hyperbolic Linear Layer . . . . .	39
4.2.5	Klein Model . . . . .	39
4.2.6	Hyperbolic Attention . . . . .	39
4.2.7	Multiclass Logistic Regression . . . . .	40
4.3	Results . . . . .	40
4.3.1	Camelyon16 . . . . .	41
4.3.2	TCGA . . . . .	43
4.4	Discussion and conclusion . . . . .	44
5	ENHANCING INSTANCE-LEVEL IMAGE CLASSIFICATION WITH SET-LEVEL LABELS . . . . .	45
5.1	Motivation . . . . .	45
5.2	Fine-Grained Representation Learning from Coarse-Grained Labels . . . . .	48
5.2.1	The FACILE Algorithm . . . . .	49
5.2.2	Theoretical Analysis . . . . .	51
5.3	Results . . . . .	53
5.3.1	Baseline Models and Algorithm Instantiation . . . . .	53
5.3.2	Pretrain with Unique Class Number of Input Sets . . . . .	54
5.3.3	Pretrain with Most Frequent Class Label . . . . .	55
5.3.4	Fine-tune CLIP Model with Anomaly Detection Dataset . . . . .	56
5.3.5	Evaluation on Histopathology Images . . . . .	58
5.3.6	Fine-tune ViT-B/14 of DINO V2 on TCGA Dataset . . . . .	61
5.4	Related Work . . . . .	61
5.5	Conclusion and Discussion . . . . .	63
6	DEEP BAYESIAN ACTIVE LEARNING . . . . .	67
6.1	Motivation . . . . .	67
6.2	Problem Setup . . . . .	70
6.2.1	Problem Statement . . . . .	70
6.2.2	Most Informative Selection Criterion . . . . .	71
6.2.3	Equivalence-class-based Selection Criterion . . . . .	73
6.3	Our Approach . . . . .	75
6.3.1	The BALanCe Acquisition Function . . . . .	75
6.3.2	Greedy Selection Strategy . . . . .	77
6.3.3	Stochastic Selection with Power Sampling and BALanCe-Clustering . . . . .	81
6.4	Experiments . . . . .	82
6.4.1	Datasets . . . . .	82
6.4.2	Experimental Setup . . . . .	84
6.4.3	Computational Complexity Analysis . . . . .	86
6.4.4	Batch-mode Deep Bayesian AL with Small Batch Size . . . . .	86
6.4.5	Effect of Different Choices of Hyperparameters . . . . .	88
6.4.6	Experiments on Tabular Datasets . . . . .	90
6.4.7	Additional Evaluation Metrics . . . . .	94

6.4.8	BALanCe via Explicit Partitioning over the Hypothesis Posterior Samples . . . . .	95
6.4.9	Batch-mode Deep Bayesian AL with Large Batch Size . . . . .	97
6.4.10	Batch-BALanCE with Multi-chain cSG-MCMC . . . . .	99
6.5	Conclusion . . . . .	100
7	CONCLUSION AND OUTLOOK . . . . .	101
7.1	Conclusion and Discussion . . . . .	101
	REFERENCES . . . . .	108
A	APPENDIX FOR ENHANCING INSTANCE-LEVEL IMAGE CLASSIFICATION WITH SET-LEVEL LABELS . . . . .	131
A.1	Training Details . . . . .	131
A.1.1	Pretrain with Unique Class Number and Most Frequent Class of Input Sets . . . . .	131
A.1.2	Fine-tune ViT-B/16 of CLIP with CUB200 . . . . .	131
A.1.3	Pretrain ResNet18 with TCGA and GTEx Dataset . . . . .	132
A.1.4	Fine-tune ViT-B/14 of DINO V2 with TCGA . . . . .	133
A.2	Additional Result . . . . .	134
A.2.1	Pretrain ResNet18 on TCGA with Patch Size 224X224 . . . . .	134
A.2.2	Benefits of Pretraining on Large Pathology Datasets . . . . .	134
A.2.3	Pretrain on TCGA and GTEx with Patch Size 1,000X1,000 . . . . .	137
A.3	Datasets . . . . .	138
A.3.1	GTEx Dataset . . . . .	138
A.3.2	TCGA Dataset . . . . .	139
A.3.3	PDAC Dataset . . . . .	141
A.3.4	NCT, PAIP, and LC . . . . .	145
A.4	Data Augmentation . . . . .	145
A.5	Latent Augmentation . . . . .	146
A.6	Ablation Study . . . . .	147
A.6.1	Set-input Models . . . . .	147
A.6.2	Learning Curve . . . . .	148
A.6.3	Input Set Size . . . . .	150
A.7	Contrastive and Non-contrastive Learning Models . . . . .	150
A.7.1	SimCLR . . . . .	151
A.7.2	SupCon . . . . .	152
A.7.3	SimSiam . . . . .	153
A.7.4	DINO and DINO V2 . . . . .	155
A.8	Excess Risk Bound of FACILE . . . . .	156
B	APPENDIX FOR DEEP BAYESIAN ACTIVE LEARNING . . . . .	164
B.1	Implementation Details on the Empirical Example in Figure. 6.1 . . . . .	164
B.2	Coefficient of Variation . . . . .	164

B.3 Predictive Variance . . . . . 165

## LIST OF FIGURES

3.1	Example tiles used for triplet loss. (a) is the anchor tile showing colonic mucosa, (b) shows predominantly colonic mucosa, and (c) shows mostly smooth muscle (from muscularis propria). (b) and (c) correspond to positive and negative samples respectively for triplet loss. . . . .	24
3.2	Example clustering visualization. The box color of each tile represents the cluster membership ( $K = 5$ ). The tile cluster colors demonstrate that tiles in a cluster are semantically coherent across and within pieces. . . . .	24
3.3	Overall architecture of PathCap. One ResNet-18 is used to extract visual features from the thumbnail of a histopathology image and pass it to the LSTM. The other ResNet-18 extracts features from randomly sampled tiles from different clusters of the histopathology image and passes them to the attention module and LSTM step by step. . . . .	26
3.4	Example tile clustering ( $K = 5$ ) with triplet loss. (a) is the original slide. (b) and (c) show the tile clustering after we train the autoencoder without and with triplet loss respectively. The colors of the boxes show the cluster membership. . . . .	27
3.5	Example slide and caption from GTEx sample GTEX-131XE-0826: <i>6 pieces; 4 pieces have full thickness elements with well preserved mucosa; 2 have no mucosa (in this section).</i> . . . . .	28
3.6	Example of visualizing caption tokens with a standard baseline model [Xu et al., 2015]. (a) and (c) are the input thumbnails to the model. (b) and (d) show the attention weights when the model generates the "myometrium" and "muscularis" tokens respectively. White/bright indicates more attention weight, black/dark indicates less attention weight. . . . .	32
4.1	Example digital hematoxylin and eosin (H&E) stained histopathology slide image with differently scaled views and the relative hierarchy. . . . .	36
5.1	(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 coarse- and fine-grained labels for histopathology images. . . . .	47
5.2	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. . . . .	49
5.3	An overview of the FACILE algorithm. (a) Pretraining step of FACILE with coarse-grained labels. The input is a set of images and the target is set-level coarse-grained label. $e$ is an instance feature map and $\phi^e$ is the corresponding set-input feature map. $g$ is the set-input model. We can instantiate the $\mathcal{A}(\ell^{cg}, \mathcal{D}_m^{cg}, \mathcal{E})$ with any supervised learning algorithms, e.g., fully supervised pretraining (FSP) with cross-entropy loss and the SupCon model. (b) Fine-grained learning of FACILE with fine-grained labels. The learned instance feature map $\hat{e}$ extracts instance-level features from patches of the support set and query set. $f$ is the fine-grained label predictor. . . . .	51



5.4	Generalization error (with two growth rates) of FACILE-FSP on CIFAR-100 test dataset as a function of the number of coarse-grained labels $m$ .	57
5.5	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 $m$ .	59
6.1	(a) The embeddings are generated by applying t-SNE on the hypotheses' predictions on a random hold-out dataset. The colorbar indicates the (approximate) test accuracy of the sampled neural networks on the MNIST dataset. See §B.1 for details of the experimental setup. (b) Probability mass (y-axis) of equivalence classes (sorted by the average accuracy of the enclosed hypotheses as the x-axis).	68
6.2	A stylized example where the most informative selection criterion underperforms the equivalence-class-based criterion.	72
6.3	Run time vs. batch size.	87
6.4	Experimental results on MNIST, Repeated-MNIST, Fashion-MNIST, EMNIST-Balanced, EMNIST-ByClass, and EMNIST-ByMerge datasets in the small-batch regime. For all plots, the $y$ -axis represents accuracy and $x$ -axis represents the number of queried examples.	88
6.5	Learning curves of different $K$ and $\tau$ for BALANCE.	89
6.6	Estimated acquisition function values $\Delta_{\text{BALANCE}}$ of BALANCE vs. posterior sample number $K$ .	90
6.7	Experimental results on 3 tabular datasets. For all plots, the $y$ -axis represents accuracy and $x$ -axis represents the number of queried examples.	91
6.8	ACC vs. # samples on the CINIC-10 dataset.	92
6.9	Performance of Random selection, BatchBALD, and Batch-BALANCE on Repeated-MNIST for an increasing number of repetitions. For all plots, the $y$ -axis represents accuracy and the $x$ -axis represents the number of queried examples. We can see that BatchBALD also performs worse as the number of repetitions is increased. Batch-BALANCE outperforms BatchBALD with large margins and remains similar performance across different numbers of repetitions.	93
6.10	ACC vs. # samples on RepeatedMNIST dataset with repeat number 3.	94
6.11	ACC vs. # samples, cSG-MCMC, CIFAR-100	95
6.12	Compare different metrics for EMNIST-Balanced and EMNIST-Bymerge	96
6.13	ACC vs. # samples for BALANCE-Partition and BALANCE.	97
6.14	Performance on SVHN and CIFAR-10 datasets in the large-batch regime.	98
6.15	ACC vs. # samples, multi-chain cSG-MCMC, CIFAR-10	99
A.1	Slide number for each organ in GTEx	139
A.2	Randomly deleted examples from GTEx dataset	140
A.3	Slide number for each tumor in TCGA	141
A.4	Randomly selected examples from TCGA dataset	142
A.5	Patch number for each tissue for PDAC	143
A.6	Randomly selected examples from each class of PDAC dataset.	144

A.7	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. . . . .	149
A.8	Abstraction of SimCLR structure . . . . .	152
A.9	Abstraction of SimSiam structure . . . . .	154
B.1	Histograms for coefficient of variation. . . . .	165
B.2	We empirically show AL algorithms' predictive variance. . . . .	166

## LIST OF TABLES

3.1	Performance on test set . . . . .	29
3.2	Influence of triplet loss . . . . .	30
3.3	Performance of PathCap with different cluster number ( $K$ ) . . . . .	31
3.4	Visualization of the PathCap method on four test slides from four different tissues. The last column shows some examples of attention weights when the model generates the corresponding tokens. White/bright indicates more attention weight, and black/dark indicates less attention weight. . . . .	33
4.1	Performance of single scale models . . . . .	42
4.2	Performance of multiple scale models . . . . .	43
4.3	Performance of single scale models . . . . .	43
4.4	Performance of multiple scale models . . . . .	44
5.1	Pretraining on input sets from CIFAR-100. Testing with 5-shot 5-way meta-test sets; average F1 and CI are reported. . . . .	55
5.2	Pretraining on input sets from CUB200. Testing with 5-shot 20-way meta-test sets; average F1 and CI are reported. . . . .	58
5.3	Test result on LC, PAIP, and NCT dataset; average F1 and CI are reported. . . . .	64
5.4	Pretraining on NCT and 5-shot 5-way testing on LC dataset; average F1 and CI are reported. . . . .	65
5.5	Test result on LC, PAIP, and NCT dataset with ViT-B/14 from DINO V2; average F1 and CI are reported. . . . .	66
6.1	Computational complexity of AL algorithms. . . . .	86
6.2	Experiment details for HAR, DRIFT and Dry Bean Dataset . . . . .	91
6.3	Mean $\pm$ STD of test accuracies when acquired training set size is 130 . . . . .	97
A.1	Models tested on LC, PAIP, and NCT dataset; average ACC and CI are reported. . . . .	135
A.2	Test result on LC, PAIP, and NCT dataset with shot number 10; average F1 and CI are reported. . . . .	136
A.3	pretraining on NCT dataset and testing on LC and PAIP dataset; average F1 and CI are reported. . . . .	137
A.4	Models pretrained on TCGA and tested on PDAC dataset; average F1 and CI are reported. . . . .	138
A.5	Models pretrained on GTEx and tested on PDAC dataset; average F1 and CI are reported. . . . .	139
A.6	Dataset statistics . . . . .	143
A.7	Performance of FACIEL-FSP with three different set-input models; average F1 and CI are reported. . . . .	149
A.8	Abation on set size; models tested on LC, PAIP, and NCT dataset; average F1 and CI are reported. . . . .	163

## ACKNOWLEDGMENTS

This dissertation signifies the culmination of an extensive and momentous journey, one that could not have been achieved without the invaluable support of numerous individuals. I wish to extend my deepest gratitude and appreciation to all those who have contributed to this scholarly work, whether through professional guidance or personal support.

First and foremost, I must express my sincere gratitude to my advisor, Prof. Robert L. Grossman, whose steadfast support and unwavering optimism over the past 6 years have been instrumental to my work. Prof. Grossman has shepherded me through the research process with remarkable patience and inspired me to think critically and maintain focus. His visionary insight, profound intellectual depth, exceptional clarity in communication, and fervent passion for research have not only shaped this dissertation but also spurred my growth as a researcher. Prof. Grossman consistently encourages enhancement in communication and presentation skills. Reflecting on the recorded sessions of our meetings, I feel immensely fortunate to have him as my advisor. His patience and enthusiasm deeply moved me. Bob, thank you! I cannot envision a more exemplary advisor.

I would like to express my sincere gratitude to the members of my dissertation committee, Prof. Aly A. Khan and Prof. Yuxin Chen, for their meticulous review of my early drafts and their many constructive comments, which have been instrumental in refining this dissertation. Prof. Aly A. Khan initially introduced me to the field of histopathology image analysis. I am profoundly grateful for his ongoing support and guidance throughout my research endeavors, internships, and job search. Prof. Yuxin Chen has been instrumental in my research, providing invaluable help with the writing and rebuttal processes of my papers. I have thoroughly enjoyed the reading groups he has conducted. Prof. Chen offers numerous beneficial suggestions regarding the presentation and structure of my thesis defense and dissertation. His optimism, deep insights, and clear logic consistently inspire me.

I extend profound appreciation to my co-authors: Robert L. Grossman, Aly A. Khan,

Yuxin Chen, Christopher R. Weber, Steven Song, Boleslaw L. Osinski, and Denise J. Lau. Collaborating with each of you has been a truly enriching experience. I am thankful for your significant contributions, which have not only enhanced this dissertation but have also had a broader impact on my academic pursuits. Your expertise and cooperation have been indispensable.

I am deeply grateful to all members of Prof. Grossman's, Prof. Khan's, and Prof. Chen's research groups, specifically Steven Song, Matthew West, Martin Putra, Derek Reiman, Hugh Yeh, Fengxue Zhang, Zixin Ding, Chaoqi Wang, Ziyu Ye, Xuefeng Liu, Zhuokai Zhao, and Yibo Jiang. Your collective contributions during numerous meetings, reading groups, dinners, and group outings have been invaluable. Your friendship and support have connected me to a dynamic academic community in this new country, enriching my experience and research journey immensely.

My sincere thanks go to the Computer Science Department, Research Computing Center, and the Center for Translational Data Science at the University of Chicago for offering an exemplary environment and computational resources that have significantly enhanced my research capabilities. Additionally, I extend my gratitude to my mentors and colleagues at Tempus Labs. Their willingness to host my internships, coupled with their valuable discussions and career guidance, has been immensely beneficial.

On a personal note, the unyielding support of my family, particularly my parents and my wife, Na Chen, has been my cornerstone. My parents have been a constant source of strength and guidance, and I owe them an immeasurable debt of gratitude. Equally, I must acknowledge my wife and best friend, Na Chen. Her compassion, understanding, and unwavering love have sustained me through both challenges and triumphs. Na, your companionship throughout this journey has deeply enriched my life, making the last six years the most treasured times of my existence. Thank you for being my partner in every sense.

In sum, this dissertation would not have been possible without the collective support and

encouragement I received, and I look forward to contributing to our field with the foundation we have all helped build.

## ABSTRACT

Diagnostic pathology and histopathology images play a critical role in the diagnosis and treatment of carcinomas. In order to achieve satisfactory performance, we usually need a large amount of labeled data. Annotating a large number of histopathology images for training machine learning models can be expensive and time-consuming. We explored several machine learning approaches in a low-data regime for histopathology images, leading to a caption generation model for histopathology images [Zhang et al., 2020b], a hyperbolic attention model for histopathology images [Zhang et al., 2020a], a deep Bayesian active learning method [Zhang et al., 2023b] to enable efficient selection of training examples that can undergo expensive annotation, and representation learning approach [Zhang et al., 2023a] that utilize existing coarse-grained labels of whole slide images to improve model performance on limited fine-grained data. Our experiments demonstrate that these approaches can improve the performances of models in the low-data regime while maintaining high levels of interpretability, minimizing labeling costs, and showing analytical advantages. The results of this study provide valuable insights for future research in the area of machine learning in low-data regimes for histopathology images.

# CHAPTER 1

## INTRODUCTION

Machine learning involves the development of algorithms that can learn from and make decisions or predictions based on data. In recent years, machine learning techniques have demonstrated remarkable success in various domains, including computer vision, natural language processing, and computational biology. This dissertation focuses specifically on the application of machine learning to healthcare, notably in histopathology image analysis. Histopathology, a crucial branch of pathology, involves the microscopic study of biological tissues to identify diseases. With advances in medical technology, digital pathology has come to the forefront. Digital pathology digitizes histopathological slides, enabling easier examination, storage, and sharing of these critical images, and has greatly enhanced the medical field.

The advent of digital pathology has introduced an abundance of complex, high-resolution data, presenting significant analysis and interpretation challenges. The traditional methods of examining these images, which often depend on the trained eyes of pathologists, are increasingly strained by the volume and complexity of the data. Machine learning can potentially facilitate this process by automating the examination of these images, leading to faster and possibly more precise diagnoses. However, this task is not straightforward; it demands complex algorithms to analyze the detailed patterns within the images, creating a computationally challenging process.

In low-data settings, the challenges become even starker due to a scarcity of labeled data. Supervised machine learning algorithms require training on labeled data, meaning in this context, histopathology images annotated with correct diagnoses. Collecting such data requires substantial time, resources, and the expertise of skilled pathologists, making it a daunting task. This dissertation aims to investigate these challenges and devise new machine learning strategies to improve the diagnostic efficiency of histopathology image analysis. The



objective is not merely about creating better algorithms but making a significant contribution to the transformation of pathology from a traditional microscope-based discipline to a digitized, data-driven one. The ultimate goal is to advance healthcare and, potentially, save lives by improving diagnostic procedures.

The first stage of our research investigates machine learning methodologies that leverage the structure of histopathology images for predicting captions and bio-markers. By leveraging the power of deep neural networks, coupled with the semantic interpretation capabilities provided by natural language processing, we aim to design a system capable of automatically generating clinically relevant captions. These captions, derived from the rich visual information present in histopathology images, provide a concise summary of vital diagnostic details. However, the scarcity of pre-existing captions and the special hierarchical structure of histopathology images make the tasks non-trivial. Furthermore, we amalgamate three core concepts—multi-scale medical image analysis, attention mechanisms, and hyperbolic embeddings into a cohesive bio-marker prediction framework. This framework underwent a thorough evaluation of two classification tasks using histopathology image datasets. The outcomes of our experiments reveal substantial enhancements in the performance of commonly utilized deep learning models.

In the subsequent stage, we explore the potential of using coarse-grained labels, such as organ-level annotations, to improve the representation learning and classification performance of our models. By leveraging the hierarchical relationships that exist between organs and their constituent tissues, we aim to provide a broader context for these images. This involves using labels that denote larger, broader categories (for example, identifying the organ from which a tissue sample originates) to understand finer, more specific details within histopathology images (such as distinguishing between healthy and diseased cells). This approach could enhance the precision of classifications at a more granular level within these complex whole-slide images.

Our research also investigates active learning, an approach that strategically selects the most informative samples for labeling. Since expert labeling is expensive in terms of time and resources, integrating active learning techniques into the training process could greatly reduce the manual effort required in data annotation, thereby improving efficiency. This approach can be particularly advantageous in the analysis of histopathology images, where achieving high performance usually requires a large volume of annotated data.

Overall, this dissertation provides an in-depth exploration of various strategies for applying machine learning to the complex field of histopathology image analysis, especially in settings where labeled data is scarce. Our goal is to expand the knowledge base and capabilities of machine learning in this domain while also contributing to practical advancements in medical diagnostics and disease management. Through this research, we aim to demonstrate the transformative potential of machine learning in healthcare, with the ultimate goal of enhancing patient care and outcomes.

## 1.1 Motivation

Learning under the low-data regime in the context of histopathology images presents several challenges. Histopathology involves the microscopic examination of tissue samples to diagnose diseases, such as cancer, based on the appearance of cells and tissues. However, due to the complex and high-dimensional nature of histopathology images, obtaining large labeled datasets for training deep learning models can be challenging. Here are some specific challenges associated with learning under the low-data regime in histopathology:

- Limited availability of labeled data: Collecting labeled histopathology images requires expert pathologists to annotate and classify each image, which is time-consuming and expensive. As a result, the number of available labeled images is often limited, making it difficult to train deep learning models effectively.

- **High-dimensional data:** Histopathology images are typically high-resolution and contain a large number of pixels. Deep learning models require a substantial amount of labeled data to learn complex patterns and features from such high-dimensional data. When the available data is limited, it becomes difficult to extract meaningful and representative features.
- **Annotation variability and subjectivity:** Histopathology image interpretation and annotation can be subjective, with different pathologists having varying opinions and expertise. Inconsistencies in annotations can introduce noise and ambiguity in the training data, making it challenging to train accurate and reliable models.

Mitigating these challenges requires the development of specialized techniques and strategies. Multi-scale models, active learning, and representation learning are some approaches that can be employed to address the limitations of low-data histopathology image learning.

## 1.2 Summary of Contributions

The primary contribution of this thesis is to investigate the machine learning methods in low data regimes for histopathology images and to propose solutions for different applications including caption generation, whole slide image classification, active learning, and representation learning with coarse-grained labels.

**Multi-scale learning for histopathology images** The analysis of histopathology images poses unique challenges due to their high-dimensional nature and the need for accurate and efficient interpretation. To address these challenges, researchers have been exploring innovative approaches that leverage deep learning techniques. We made two contributions: the development of caption generation models specifically designed for histopathology whole-slide images and the integration of hyperbolic attention models into the classification

of histopathology images. These advancements offer novel perspectives and demonstrate promising results in enhancing the understanding and analysis of histopathology images.

- Caption generation from histopathology images: The automatic generation of captions from medical images offers an efficient solution for annotating histopathology images, facilitating image retrieval tasks, and promoting the standardization of clinical ontologies. In this study, our focus lies on the development and methodical evaluation of a novel caption generation framework specifically designed for histopathology whole-slide images. Introducing PathCap, a deep learning multi-scale framework, we leverage multi-scale views of whole-slide images to predict accurate and informative captions. Through comprehensive evaluations, we demonstrate the superior performance of our framework compared to a standard baseline model across a diverse range of human tissues. Furthermore, our approach provides interpretable contextual cues that enhance the understanding of generated captions. Additionally, we present a novel dataset of histopathology images with captions sourced from the Genotype-Tissue Expression (GTEx) project. This dataset serves as a valuable resource for the machine learning and healthcare community, enabling benchmarking of future caption prediction and interpretation methods. The reference code for our work is publicly available at <https://github.com/zhangrenyuuchicago/PathCap>.
- Hyperbolic attention model for histopathology image classification: Our work integrates three fundamental concepts—multi-scale medical image analysis, attention mechanisms, and hyperbolic embeddings—into a unified framework. Notably, the formulation and evaluation of hyperbolic attention models for multi-scale medical image analysis have remained unexplored until now. In this paper, we present a comprehensive evaluation of a hyperbolic attention model on two classification tasks using histopathology image datasets. Our experiments demonstrate significant improvements compared to commonly used deep learning models. By directly capturing the multi-

scale structure of histopathology images, our method effectively highlights one or more discriminative structures at various scales, facilitated by the inherent nature of the hyperbolic attention mechanism. We release the reference code for our approach, which is available at <https://github.com/zhangrenyuuchicago/PathHyperbolic>.

Both the caption generation model, PathCap, and the hyperbolic attention model have demonstrated notable performance improvements and enhanced interpretability in the analysis of histopathology images. PathCap outperforms standard baseline models in accurately predicting informative captions, while the hyperbolic attention model effectively highlights discriminative structures at various scales, leading to significant improvements over commonly used deep learning models. These advancements not only contribute to the field of histopathology image analysis but also provide valuable insights and tools for improved understanding and interpretation of complex medical image data.

**Deep Bayesian active learning approach for data acquirement** In the context of histopathology image analysis, where labeled data is often limited, researchers have been exploring various methods to enhance model performance. One such method is active learning, which offers a smarter approach to selecting samples for expert annotation, resulting in improved performance while minimizing the query cost. Active learning has demonstrated its data efficiency across numerous fields, including histopathology. Existing active learning algorithms, especially in the context of batch-mode deep Bayesian active models, rely heavily on the quality of uncertainty estimations of the model and are often challenging to scale to large batches. In this paper, we propose Batch-BALANCE, a scalable batch-mode active learning algorithm, which combines insights from decision-theoretic active learning, combinatorial information measure, and diversity sampling. At its core, Batch-BALANCE relies on a novel decision-theoretic acquisition function that facilitates differentiation among different *equivalence classes*. Intuitively, each equivalence class consists of hypotheses (e.g.,

posterior samples of deep neural networks) with similar predictions, and Batch-BALANCE adaptively adjusts the size of the equivalence classes as learning progresses. To scale up the computation of queries to large batches, we further propose an efficient batch-mode acquisition procedure, which aims to maximize a novel *information measure* defined through the acquisition function. We show that our algorithm can effectively handle realistic multi-class classification tasks and achieves compelling performance on several benchmark datasets for active learning under both low- and large-batch regimes. Reference code is released at <https://github.com/zhangrenyuuchicago/BALanCe>.

### **Representation learning with coarse-grained labels for histopathology images**

Classifying clinical properties directly from histopathology images is a vital step toward improving and augmenting processes in clinical and healthcare settings. A large number of labels for histopathology images are needed for models to get a good performance. To address this issue, we introduce a novel representation learning approach designed to leverage the hierarchical relationship between organs, which bear *coarse-grained labels*, and tissues, which carry *fine-grained labels*. The coarse-grained labels are easier to get than the fine-grained labels and there are many publicly available coarse-grained labels for histopathology images. The proposed few-shot algorithm, requiring only a handful of fine-grained annotated samples, learns representations that enable proficient fine-grained label predictions. Empirical evaluations conducted across diverse histopathology image datasets demonstrate the algorithm’s efficacy, with the model exhibiting superior performance in comparison to established pretraining and self-supervised learning techniques. A theoretical analysis of the algorithm is also presented. These findings suggest that our approach may provide a promising avenue for the development of efficient learning models for histopathology images, even in the presence of limited fine-grained data. We released our reference code in <https://github.com/zhangrenyuuchicago/FACILE>.

## 1.3 Thesis Organization

This thesis is organized into five chapters, each focusing on a specific aspect of machine learning in the low-data regime for histopathology images. The chapters are structured as follows:

**Chapter 1: Introduction** The first chapter provides an introduction to the research topic, outlining the motivation, objectives, and research questions addressed in the thesis. It highlights the significance of machine learning in the context of histopathology image analysis and sets the foundation for the subsequent chapters.

**Chapter 2: Background** The second chapter presents a comprehensive background on the relevant theories, methodologies, and techniques in machine learning for histopathology image analysis. It reviews the existing literature on low-data regime approaches, active learning, caption generation, and learning with coarse-grained labels. This chapter establishes the theoretical framework for the subsequent chapters and provides a solid understanding of the key concepts.

**Chapter 3: Caption Generation** Chapter 3 delves into the topic of caption generation for histopathology images. It explores the different approaches, such as deep learning models and natural language processing techniques, used to generate informative and accurate captions from histopathology images. The chapter discusses the challenges specific to histopathology images and presents novel methods and algorithms to address them.

**Chapter 4:** This chapter discusses the integration of deep learning in computer vision, focusing on histopathology imaging and the limitations of traditional Euclidean embeddings for complex hierarchical data. It introduces an innovative approach using hyperbolic spaces to improve the handling and classification of large-scale histopathology images. The text

highlights the potential of this method to enhance histopathology image analysis by effectively combining different magnifications and details within a hyperbolic framework.

**Chapter 5: Learning with Coarse-Grained Labels** Chapter 5 focuses on learning with coarse-grained labels in histopathology image analysis. It investigates the use of easily accessible coarse-grained annotations, such as organ-level labels, to enhance representation learning and improve the classification and analysis of histopathology images. The chapter presents novel methodologies that leverage the hierarchical relationships between organs and tissues to facilitate learning in the low-data regime.

**Chapter 6: Active Learning** Chapter 6 explores the use of active learning techniques in histopathology image analysis. It discusses strategies to intelligently select the most informative samples for annotation, thereby maximizing the effectiveness of the limited labeled data available. The chapter investigates the integration of active learning with machine learning models for histopathology image classification and other tasks, highlighting the benefits and challenges associated with this approach.

**Chapter 7: Conclusion and Future Directions** Following the main chapters, the thesis concludes with a summary of the key findings, contributions, and implications of the research. It also highlights potential avenues for future research in machine learning for histopathology image analysis in the low-data regime.

## 1.4 List of Publications

This dissertation is based on materials from our following conference publications:

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



- 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), pages 773–776. IEEE, 2020a.
- Renyu Zhang, Aly A Khan, Robert L Grossman, and Yuxin Chen. Scalable batch-mode deep Bayesian active learning via equivalence class annealing. In The Eleventh International Conference on Learning Representations (ICLR), 2023.
- Renyu Zhang, Aly A Khan, Yuxin Chen, and Robert L Grossman. Enhancing Instance-Level Image Classification with Set-Level Labels. In The Twelfth International Conference on Learning Representations (ICLR), 2024.

## CHAPTER 2

### BACKGROUND AND RELATED WORKS

#### 2.1 Histopathology Images

We provide an overview of the processes involved in tissue preparation, staining, and digitization of histological slides. In the typical workflow at a hospital, tumor excisions or biopsies are performed in the operating room, and the collected material is then sent to the pathology lab for analysis.

The initial step in the tissue preparation process involves formalin fixation and embedding in paraffin. The tissue samples are immersed in formalin to ensure preservation and then embedded in paraffin blocks. To create thin sections for analysis, a microtome, which is a precise cutting instrument, is used to cut sections with a thickness of  $3 - 5\mu\text{m}$ . These sections are then mounted on glass slides.

Although the structures of interest in the tissue, such as nuclei and cytoplasm, are not easily visible on the mounted sections, they can be highlighted through staining. The standard staining protocol involves the use of hematoxylin and eosin (H&E). Despite being in use for nearly a century, H&E staining remains the primary diagnostic and prognostic procedure for most patients. Hematoxylin binds to DNA, resulting in a blue/purple coloration of the nuclei, while eosin binds to proteins, dyeing other structures such as the cytoplasm and stroma pink.

By employing the H&E staining technique, pathologists can identify and analyze cellular structures, aiding in the diagnosis and characterization of tissue samples. It serves as a fundamental step in histopathology analysis, providing valuable information for medical professionals.

In addition to tissue preparation and staining, digitization of histological slides has become increasingly important. The process involves capturing high-resolution images of

stained slides using digital scanners. This digitization allows for convenient storage, sharing, and analysis of histopathology images, facilitating remote consultations, research collaborations, and computer-aided diagnostic systems.

## 2.2 Caption Prediction

Early image caption generation focused on detection [Kulkarni et al., 2013] followed by template filling. Since the rise of deep learning, most caption generation models have adopted the encoder-and-decoder paradigm [Vinyals et al., 2015b, Xu et al., 2015]. These methods typically use non-medical images, such as nature scenes found in ImageNet [Russakovsky et al., 2015b]. Typically, the encoder is a CNN that extracts features from input images, and the decoder uses an LSTM [Hochreiter and Schmidhuber, 1997] to generate tokens step by step. Notably, Xu et al. [2015] incorporated the attention mechanism into the encoder-and-decoder paradigm by feeding an attention-weighted combination of features to the LSTM. This approach has proven to be very effective in terms of performance and now defines the standard baseline caption model. However, the visualization and interpretation of the attention weight on the input images can be very ambiguous and non-specific. Subsequent work in this field focused on further exploiting attention, e.g., You et al. [2016] plugged the attention-weighted features over semantic concepts into hidden states of LSTM and words generation layers, and Liu et al. [2017a] proposed to use instance segmentation to improve the correctness of attention.

More closely related to medical imaging, Zhang et al. [2017] was aimed at generating semi-structured pathology descriptions. In order to gain effective gradient flow for training, they utilized a predefined subset of descriptions extracted from the reports. They demonstrated slightly better performance in their experiments compared to a standard baseline caption model. Jing et al. [2017] also adopted an encoder-and-decoder paradigm for X-ray images and developed a hierarchical LSTM model to specifically overcome the challenges of long

paragraphs in clinical reports.

Collectively, these methods all require non-trivial changes to adapt to histopathology images due to the lack of instance segmentation information in routine imaging data and robust clinical pathology instance detectors. Furthermore, these methods require rescaling whole-slide images for implementation, causing loss of high-resolution information about the sample tissue and morphology and thus, limiting their ability to utilize full-resolution data for generating salient captions.

## 2.3 Active Learning

Active learning (AL) [Settles, 2012] characterizes a collection of techniques that efficiently acquire data for training machine learning models. In the pool-based setting, an active learner selectively queries the labels of data points from a pool of unlabeled examples and incurs a certain cost for each label obtained. The goal is to minimize the total cost while achieving a target level of performance. A common practice for AL is to devise efficient surrogates, also known as acquisition functions, to assess the effectiveness of unlabeled data points in the pool. In our study, we designed a novel acquisition function to better suit Bayesian neural networks and data characteristics, leading to more cost-effective learning. Additionally, the methodologies reviewed here influenced the design of our data selection strategies.

There has been a vast body of literature and empirical studies [Huang et al., 2010, Hounsby et al., 2011, Wang and Ye, 2015, Huang et al., 2016, Sener and Savarese, 2017, Ducoffe and Precioso, 2018, Ash et al., 2019, Liu et al., 2020, Yan et al., 2020] suggesting a variety of heuristics as potential acquisition functions for AL. Among these methods, Bayesian Active Learning by Disagreement (BALD) [Hounsby et al., 2011] has attained notable success in the context of deep Bayesian AL, while maintaining the expressiveness of Bayesian models [Gal et al., 2017, Janz et al., 2017, Shen et al., 2017]. Concretely, BALD relies on a most

informative selection (MIS) strategy—a classical heuristic that dates back to Lindley [1956]—which greedily queries the data point exhibiting the maximal mutual information with the model parameters at each iteration. Despite the overwhelming popularity of such heuristics due to the algorithmic simplicity [MacKay, 1992, Chen et al., 2015c], the performance of these AL algorithms unfortunately is sensitive to the quality of uncertainty estimations of the underlying model, and it remains an open problem in deep AL to accurately quantify the model uncertainty, due to limited access to training data and the challenge of posterior estimation.

**Bayesian active learning** Batch-mode AL has shown promising performance for practical AL tasks. Recent works, including both Bayesian [Houlsby et al., 2011, Gal et al., 2017, Kirsch et al., 2019] and non-Bayesian approaches [Sener and Savarese, 2017, Ash et al., 2019, Citovsky et al., 2021, Kothawade et al., 2021, Hacohen et al., 2022], have been enormous and we hardly do it justice here. We mentioned what we believe is the most relevant in the following. Among the Bayesian algorithms, Gal et al. [2017] choose a batch of samples with top acquisition functions. These methods can potentially suffer from choosing similar and redundant samples inside each batch. Kirsch et al. [2019] extended Houlsby et al. [2011] and proposed a batch-mode deep Bayesian AL algorithm, namely BatchBALD. Chen and Krause [2013b] formalized a class of interactive optimization problems as adaptive submodular optimization problems and proved a greedy batch-mode approach to these problems is near-optimal as compared to the optimal batch selection policy. ELR focuses on a Bayesian estimate of the reduction in classification error and takes a one-step-look-ahead strategy [Roy and McCallum, 2001]. Inspired by ELR, WMOCU [Zhao et al., 2021] extends MOCU [Yoon et al., 2013] with a theoretical guarantee of convergence. However, none of these algorithms extend to the batch setting.

**Non-Bayesian active learning** Among the non-Bayesian approaches, Sener and Savarese [2017] proposed a CoreSet approach to select a subset of representative points as a batch. BADGE [Ash et al., 2019] selects samples by using the k-MEANS++ seeding algorithm from the AL pool, which are the gradient embeddings of DNN’s last layer induced by hallucinated labels. Contemporary works propose AL algorithms that work for different settings including text classification [Tan et al., 2021], domain shift and outlier [Kirsch et al., 2021b], low-budget regime [Hacohen et al., 2022], very large batches (e.g., 100K or 1M) [Citovsky et al., 2021], rare classes, and OOD data [Kothawade et al., 2021].

## 2.4 Few-shot Learning

In recent years, few-shot learning (FSL) has gained popularity as a method for adapting models to new tasks with limited labeled data [Vinyals et al., 2016, Finn et al., 2017, Wang and Hebert, 2016, Triantafillou 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., 2020, Lee et al., 2019b, Li et al., 2019]. In traditional supervised learning approaches, a large amount of labeled data is typically required to learn a robust model. However, in real-world scenarios, obtaining enough labeled data for each task is frequently difficult or impossible. FSL addresses this by employing techniques that allow a model to generalize from a few data points to new tasks and classes not seen during training, simulating a more human-like rapid learning ability.

The categorization of FSL methods is often grouped into three main categories based on their underlying principles and methodologies: metric learning, model-based Learning, and optimization-based Learning. Each of these categories utilizes different strategies to tackle the small sample size problem inherent in FSL.

Metric learning methods focus on learning a similarity function that maps inputs to an embedding space where the distance between similar items is minimized, and the distance between dissimilar items is maximized. Siamese Networks [Koch et al., 2015] utilize a twin

network architecture that learns to differentiate between pairs of inputs, training on whether pairs are similar or not. Prototypical Networks [Snell et al., 2017] learn a metric space in which classification can be performed by computing distances to prototype representations of each class that are the mean of the embedded points belonging to that class. Matching Networks [Vinyals et al., 2016] frame the FSL problem as a weighted nearest neighbor classification, updated by an attention mechanism over a learned embedding.

Model-based methods attempt to learn a predictive model that can quickly adapt to new tasks with minimal data. Memory-augmented neural networks [Santoro et al., 2016] incorporate external memory components to rapidly assimilate new data and make predictions with them, facilitating fast learning and adaptation. Meta Networks [Munkhdalai and Yu, 2017] implement meta-learning through fast parameterization and slow learning weights, enabling quick adaptation to new tasks.

Optimization-based learning methods are designed to optimize the model parameters effectively for fast learning on new tasks, typically through the initialization of model parameters that are particularly adaptable. Model-Agnostic Meta-Learning (MAML) [Finn et al., 2017] optimizes a model’s initial parameters so that a small number of gradient updates will lead to good performance on a new task. Reptile [Nichol and Schulman, 2018] is a simplification of MAML that performs stochastic gradient descent on a small number of tasks and moves the initialization towards the weights that perform well on these tasks.

Self-supervised learning has recently emerged as a promising approach for FSL by leveraging unlabeled data to learn useful representations without the need for extensively annotated datasets [Brown et al., 2020, Lu et al., 2022]. This method is especially valuable in scenarios where labeled data is scarce or costly to acquire. In the realm of image classification, Gidaris et al. [2019], Mangla et al. [2020], Su et al. [2020] utilize the pretext tasks of self-supervised learning as an auxiliary loss to enhance the representation learning of supervised pretraining. However, the performance of these methods significantly degrades without supervision. An-

other approach involves unsupervised FSL, where Antoniou and Storkey [2019], Hsu et al. [2018], Khodadadeh et al. [2019, 2020], Lee et al. [2020], Medina et al. [2020], Qin et al. [2020], Lu et al. [2022] adapt existing supervised meta-learning methods to unsupervised versions. Additionally, similar efforts in continual [Gallardo et al., 2021] and open-world learning [Dhamija et al., 2021] also benefit from self-supervised learning to enhance their performance.

FSL in medical images is still at its early stage [Yang et al., 2022]. Mahajan et al. [2020] proposed an FSL method named Meta-DermDiagnosis for skin lesion datasets. Chen et al. [2021c] used momentum contrastive learning [He et al., 2020] to train an encoder with a large and publicly available lung dataset and adopt the prototypical network [Snell et al., 2017] for classification. Medela et al. [2019] proposed to train VGG16 [Simonyan and Zisserman, 2014] with a non-linear version of triplet loss [Schroff et al., 2015] and fine-tune a SVM to test on new tasks. Sikaroudi et al. [2020] explored the performance of DNN and triplet loss [Schroff et al., 2015] in the area of representation learning and applied FSL to two publicly available datasets: The Cancer Genome Atlas (TCGA) and colorectal cancer (CRC) dataset [Kather et al., 2016]. Teh and Taylor [2020] showed that features learned from weakly labeled dataset, i.e., KimiaPath24 [Babaie et al., 2017], are transferable and allow us to achieve highly competitive path classification results on CRC dataset [Kather et al., 2016] and PatchCamelyon (PCam) dataset [Veeling et al., 2018] while using an order of magnitude less labeled data. Sikaroudi et al. [2020] proposed a benchmark for the few-shot classification of histology images. Yang et al. [2022] conducted investigations on more settings including generalized few-shot learning and hetero-/homo-geneous few-shot selection.

## 2.5 Self-supervised Learning

Self-supervised learning (SSL) has emerged as a promising approach in machine learning, particularly in the field of computer vision, for learning representations without the need



for explicit labels. We follow the survey of Uelwer et al. [2023] and categorize SSL methods into five main groups: pretext task methods, information maximization methods, clustering-based methods, contrastive learning methods, and teacher-student methods. In our research, we employ and compare with SSL methods, which have proven effective in previous studies for histopathology images, to train our few-shot learning model [Yang et al., 2022]. This choice was influenced by their demonstrated ability to enhance feature discriminability with minimal labeled data. Furthermore, we integrate insights from the latent augmentation method [Yang et al., 2022] to improve the robustness and generalizability of our model.

Pretext task methods revolve around the idea of creating a supervised learning scenario wherein the model is trained to predict artificially created labels. Tasks such as predicting the rotation of an image [Gidaris et al., 2018], solving jigsaw puzzles, or reconstructing images [Le Cun and Fogelman-Soulié, 1987, Vincent et al., 2010, Rifai et al., 2011, Kingma and Welling, 2013] are typical examples. These tasks encourage the model to focus on the inherent structure of the data necessary for performing these tasks, thereby learning useful features for downstream applications. For instance, the Rotation Network encourages learning features that are orientation-invariant, useful for tasks where orientation changes but the semantic content does not.

Information maximization methods aim to learn representations that are invariant to input transformations while maximizing the information content in the learned features. Approaches like Barlow Twins [Zbontar et al., 2021] and VICReg [Bardes et al., 2021] work by reducing redundancy among the features, thus preventing the collapse of representations (where different inputs might produce the same output features). These methods often employ statistical constraints such as cross-correlation matrices between different augmented views of the same image to ensure the learned features are both diverse and rich in information.

Clustering-based methods, such as DeepCluster [Caron et al., 2018] and SwAV [Caron

et al., 2020], utilize unsupervised clustering techniques to group the data into clusters, which are then used as pseudo-labels for training. This approach not only helps in learning features that are invariant to input modifications (as the same object in different orientations should ideally belong to the same cluster) but also aligns closely with how humans categorize objects in the real world—based on their overall similarity rather than specific labels.

Contrastive learning methods, like SimCLR [Chen et al., 2020a,b] and MoCo [He et al., 2020, Chen et al., 2020c, 2021d], leverage the contrast between similar (positive) and dissimilar (negative) examples to guide the learning process. By pushing apart dissimilar examples and pulling together similar ones, these methods encourage the model to learn generalizable features that robustly categorize the data. The effectiveness of these methods often hinges on the choice of positive and negative samples, the transformation strategies used to generate these samples, and the architectural details like the use of projection heads.

Teacher-student methods such as BYOL [Grill et al., 2020] and SimSiam [Chen and He, 2021] involve pairs of networks where the 'student' learns to predict the output of the 'teacher'. This setup is beneficial as it stabilizes the learning process—the teacher gradually evolves during training, providing a moving target for the student's predictions, which prevents overfitting and helps the student explore a richer set of features. Notably, these methods do not require negative pairs (common in contrastive learning), simplifying training and reducing the potential for representational collapse.

These SSL methods have shown impressive performance in learning powerful representations from unlabeled data. They have been successfully applied to various computer vision tasks, such as image classification, object detection, and semantic segmentation. The effectiveness of SSL has opened up new possibilities for learning from unannotated data and reducing the dependency on large labeled datasets.

## CHAPTER 3

# CAPTION GENERATION FOR HISTOPATHOLOGY IMAGES

### 3.1 Motivation

In the past decades, significant advancements in clinical pathology, such as biospecimen fixation, staining, and digital microscopy, have revolutionized the field by enabling the routine digitization of histopathology slides [Bera et al., 2019]. Histopathological images contain a wealth of clinical diagnostic information. For instance, in colonic biopsies, they offer insights into architectural details, including crypt abnormalities and the distribution of inflammatory cells, which provide valuable insights into disease processes. Anatomic pathologists have developed specialized language and lexicons to effectively communicate these descriptive findings. However, automatically describing the content of histopathology images poses a grand challenge in machine learning, as it requires the integration of computer vision and natural language processing disciplines. Accurate machine learning methods capable of generating and visualizing captions from histopathology images have the potential to revolutionize various applications. Firstly, they can support pathologists by providing caption prompts and visual cues to facilitate clinical review. Secondly, they can enable image retrieval tasks, such as searching for specific labels or descriptions in archival histopathology slide images. The development of such methods holds tremendous promise for enhancing the practice of histopathology and expanding its applications in healthcare.

The precise characterization of fine-grained morphological and pathological features that distinguish various classifications in histopathology traditionally relies on expert visual assessment, necessitating years of training and honing visual skills [Brugnara et al., 1994]. Surprisingly, the application of machine learning techniques for automatically generating natural language descriptions from histopathology images has received limited attention, and the availability of benchmark datasets for histopathology caption prediction tasks remains

inadequate. Motivated by this gap, our study aims to methodically evaluate the feasibility of generating short, clinically relevant descriptions (captions) from H&E histopathology whole-slide images using automated methods. Furthermore, we contribute a benchmark dataset tailored for the machine learning community, promoting advancements in this domain and encouraging further research in histopathology caption generation.

Deep neural networks have demonstrated remarkable success in various complex tasks involving histopathology images, such as tissue classification [Bejnordi et al., 2017], disease outcome prediction [Mobadersany et al., 2018], and genetic alteration prediction [Coudray et al., 2018]. These networks can directly learn fine-grained features from raw images in supervised learning settings. However, applying standard machine learning techniques to caption prediction in histopathology faces significant challenges. Firstly, histopathology images often exceed one billion pixels (gigapixel), posing memory limitations for off-the-shelf deep neural network models. Rescaling high-resolution images to circumvent memory constraints can lead to the loss of crucial contextual and spatial information, hindering the generation of relevant descriptions from whole-slide histopathology images. Secondly, there is a need for methods to evaluate and visually interpret the generated captions, enabling their effective adoption in clinical practice. Consequently, the joint task of predicting and interpreting captions in the context of gigapixel-sized images presents a technically demanding problem that is largely unique to the healthcare domain.

We present PathCap, a novel multi-scale view framework designed for histopathology whole-slide images. PathCap employs a two-step approach, starting with the clustering of high-resolution tiles extracted from the images. Subsequently, it combines a single low-resolution thumbnail view of the whole-slide image with randomly sampled tiles from the high-resolution clusters. Through our experiments, we demonstrate that PathCap effectively leverages and integrates information from both high-resolution and low-resolution views. To evaluate our framework, we conducted tests on data obtained from the Genotype-Tissue

Expression (GTEx) project [Lonsdale et al., 2013]. Additionally, we collaborated with a pathologist to evaluate our caption predictions, enabling us to identify both the limitations and opportunities associated with caption prediction from histopathology images. Parts of this chapter are replicated from Zhang et al. [2020b] with some modifications.

## 3.2 Methods

### 3.2.1 Overview

The tissue regions within H&E whole-slide images  $\{s^i\}_{i=1}^M$  are tiled into non-overlapping sections (1000x1000px)  $\{t_j^i\}_{j=1}^{N^i}$ . Here  $M$  is the number of whole-slide images in the dataset. The  $N^i$  is the number of tiles that contain tissues and are extracted from slide  $s^i$ . The tissue region is deduced by selecting tiles with an average grayscale pixel value in the range [0.2, 0.7]. An autoencoder is trained on tissue containing tiles  $\{t_j^i\}_{j=1}^{N^i}$  using both reconstruction loss and triplet loss. We cluster the tiles  $\{t_j^i\}_{j=1}^{N^i}$  extracted from each slide  $s^i$  based on the embeddings  $\{e_j^i\}_{j=1}^{N^i}$  learned from the autoencoder. For simplicity, we focus our study on k-means, but other clustering approaches can be used as well. K-means takes a set of vectors as input, in our case the embedding produced by an autoencoder, and clusters  $\{e_j^i\}_{j=1}^{N^i}$  into  $K$  distinct groups  $\{C_k^i\}_{k=1}^K$  based on a Euclidean distance. Thus, if we fix the cluster number  $K$  as 5, the tiles from tissue regions in each histopathology image are clustered into 5 groups.

Next, a rescaled thumbnail  $b^i$  and tiles  $\{t_k^i\}_{k=1}^K$  sampled from each cluster  $\{C_k^i\}_{k=1}^K$  of a slide  $s^i$  are fed into our caption generation model (PathCap) during training and testing. If the cluster number is set to  $K = 5$ , five tiles, 1 from each cluster, are sampled randomly. The thumbnail  $b^i$  is used to initialize the LSTM, and tiles  $\{t_k^i\}_{k=1}^K$  are fed to the LSTM step by step. Our attention module is based on the sampled tiles. To enable visualization of the attention across the whole slide image, we can show the attention weights over all tiles  $\{t_k^i\}_{k=1}^K$  from a given cluster  $\{C_k^i\}_{k=1}^K$ . We used PyTorch to implement our model

Vinodababu [2019].

### 3.2.2 Metric Learning with Triplet Loss

A key step in PathCap involves clustering semantically similar high-resolution tiles from histopathology whole-slide images. In order to cluster tiles within a whole-slide image, we sought to learn embeddings for arbitrary image tiles such that similar tiles have similar embeddings. To accomplish this we used metric learning, which aims to produce a feature space  $\mathcal{F}$  with a certain metric structure, where similarity can be captured by some distance function, typically the Euclidean distance [Ho et al., 2019]. In the context of deep learning, classic metric learning uses no additional layers. Several variants have been proposed [Movshovitz-Attias et al., 2017], [Sohn, 2016], [Hadsell et al., 2006], and [Chopra et al., 2005]; among which triplet loss [Schroff et al., 2015], [Wang et al., 2014], [Bell and Bala, 2015], and [Weinberger and Saul, 2009] is the most popular. In practice, triplets make the training difficult by increasing the sample number cubically. Many methods have sought to accelerate the training [Wang et al., 2014], [Bell and Bala, 2015], [Schroff et al., 2015], and [Oh Song et al., 2016].

An autoencoder is trained on all tissue containing tiles  $\{t_j^i\}$  extracted from all slides  $\{s^i\}_{i=1}^M$  in the dataset. An autoencoder is an unsupervised method that generates a small compressed feature representation or embedding for each input sample. These features can capture the variance of the whole dataset while exhibiting a small amount of reconstruction loss. The large amount of tiles extracted from gigapixel histopathology slides makes it computationally expensive to process all the tiles from a slide within one single pass. Instead, we randomly sample a limited number of tiles for each slide.

To learn a more robust embedding, in addition to the reconstruction loss, we use triplet loss. Specifically, during the training of the autoencoder, the data loader returns a set of triplet  $(t_j^i, t_k^i, t_l^i)$  tiles from each slide  $s^i$ .  $t_j^i$  is the anchor tile.  $t_k^i$  is a positive example of  $t_j^i$ .

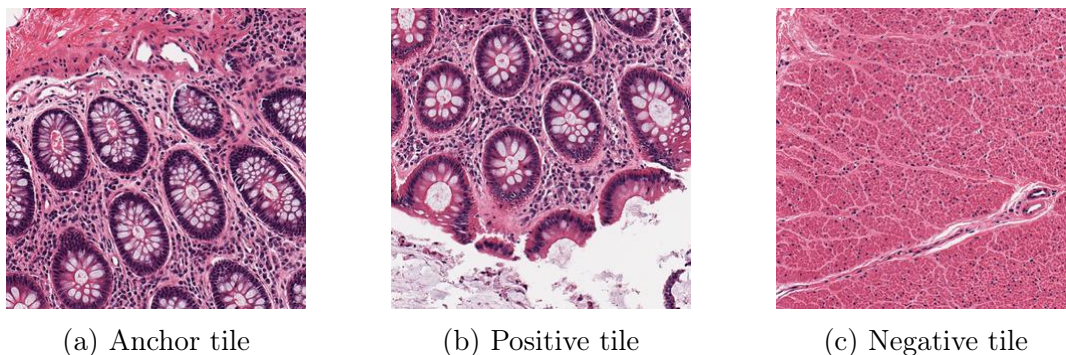


Figure 3.1: Example tiles used for triplet loss. (a) is the anchor tile showing colonic mucosa, (b) shows predominantly colonic mucosa, and (c) shows mostly smooth muscle (from muscularis propria). (b) and (c) correspond to positive and negative samples respectively for triplet loss.

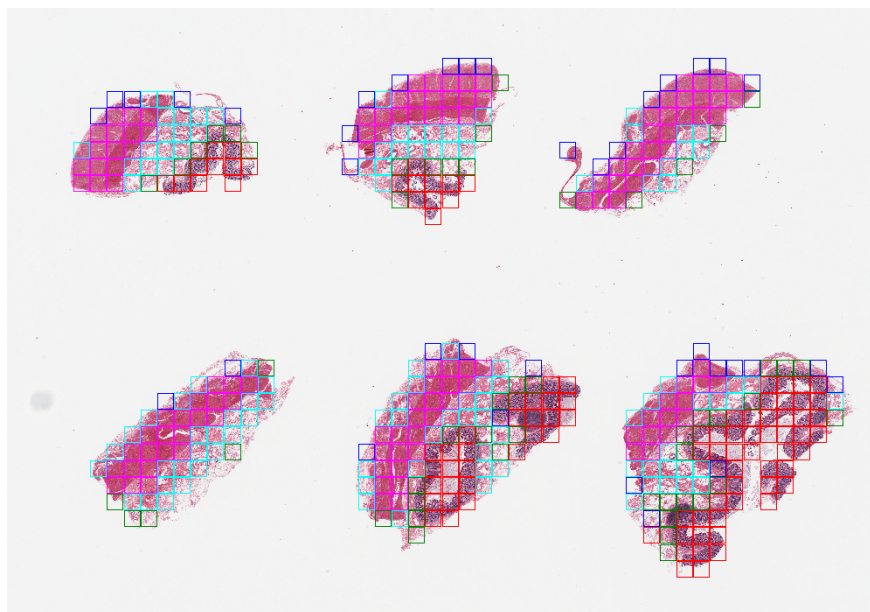


Figure 3.2: Example clustering visualization. The box color of each tile represents the cluster membership ( $K = 5$ ). The tile cluster colors demonstrate that tiles in a cluster are semantically coherent across and within pieces.

Here we define positive examples as an adjacent tile.  $t_l^i$  is a negative example of  $t_j^i$ , which means  $t_l^i$  is not adjacent to  $t_j^i$ . Tile examples can be seen in Figure. 3.1. The loss is as follows:

$$L(t_j^i, t_k^i, t_l^i) = \mu \cdot \max(d(e_j^i, e_k^i) - d(e_j^i, e_l^i) + m, 0) + d(t_j^i, D(e_j^i))$$

$E$  is encoder and  $D$  is decoder.  $e_j^i = E(t_j^i)$ .  $d(\cdot, \cdot)$  represents the distance.  $m$  is the margin and  $\mu$  is the factor for triplet loss. We use mean squared deviation as the distance.

We train the autoencoder with the Adam method [Kingma and Ba, 2014]. The autoencoder is trained for 4 epochs. After the training of the autoencoder is finished, we use the autoencoder to obtain representations for all the tiles. For each slide, we perform k-means clustering for all the tiles in the slide. Clustering example is shown in Figure. 3.2.

### 3.2.3 Neural Network Architecture

Low-resolution thumbnail images are used to initialize the LSTM [Hochreiter and Schmidhuber, 1997]. An attention mechanism on tiles is adopted for each step of generating captions, following the approach from Ilse et al. [2018]. Overall, PathCap contains three modules (Figure. 3.3): the thumbnail encoder, tiles encoder, and decoder.

For the thumbnail encoder part, the standard ResNet-18 [He et al., 2016] extracts the feature vector from a given input image thumbnail  $b^i$ . The feature vector is linearly transformed and then used to initialize LSTM.

The tile encoder contains another ResNet-18 to extract representations from tiles  $\{t_k^i\}_{k=1}^K$ . Let  $H^i = \{h_k^i\}_{k=1}^K$  be a bag of  $K$  representations of  $K$  tiles from different clusters  $\{C_k^i\}_{k=1}^K$  of a slide  $s^i$ . The attention-weighted representation  $z^t$  at step  $t$  for a slide  $s^i$  is

$$z^t = \sum_{k=1}^K \alpha_k^t h_k$$



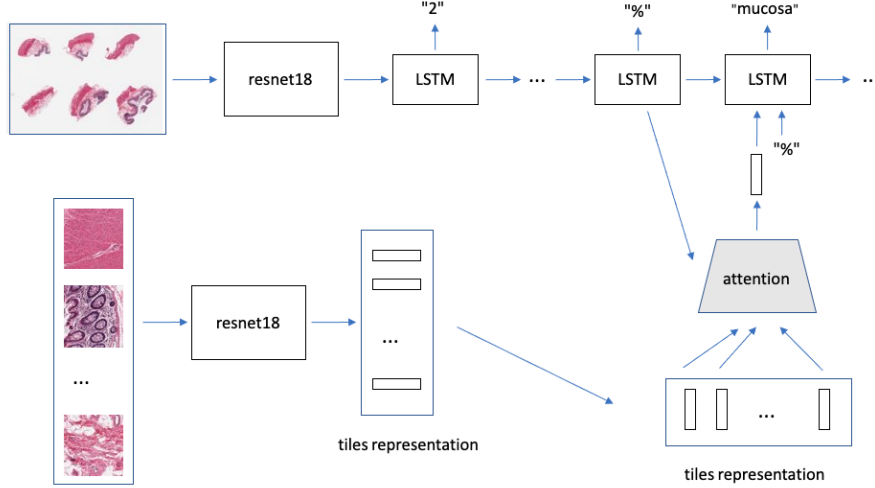


Figure 3.3: Overall architecture of PathCap. One ResNet-18 is used to extract visual features from the thumbnail of a histopathology image and pass it to the LSTM. The other ResNet-18 extracts features from randomly sampled tiles from different clusters of the histopathology image and passes them to the attention module and LSTM step by step.

where:

$$\alpha_k^t = \frac{\exp(w^T \tanh(V[h_k, m^t]))}{\sum_{g=1}^K \exp(w^T \tanh(V[h_g, m^t]))}$$

$m^t$  is the hidden state of LSTM at step  $t$ , and  $w$  and  $V$  are parameters of two linear layers.  $[\cdot, \cdot]$  is the concatenation operation.

For the decoder part of PathCap, source and target texts are predefined. For example, if the image description is "2 pieces, 15% vessel stroma, rep delineated", the source sequence is a list containing [ $\langle \text{start} \rangle$ , '2', 'pieces', ',', '15%', 'vessel', 'stroma', 'rep', 'delineated'] and the target sequence is a list containing ['2', 'pieces', ',', '15%', 'vessel', 'stroma', 'rep', 'delineated',  $\langle \text{end} \rangle$ ]. Using these source and target sequences and the feature vector, the LSTM decoder is trained as a language model conditioned on the image feature vector. Notably, we can use the attention mechanism to extract features from sampled tiles and visualize the weights when generating each word of a caption for histology images.

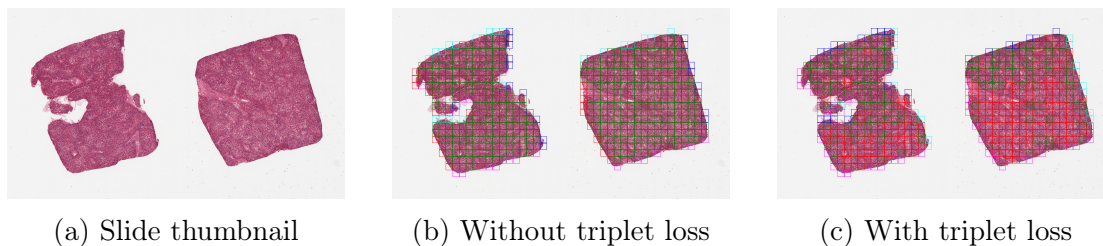


Figure 3.4: Example tile clustering ( $K = 5$ ) with triplet loss. (a) is the original slide. (b) and (c) show the tile clustering after we train the autoencoder without and with triplet loss respectively. The colors of the boxes show the cluster membership.

### 3.2.4 Data Augmentation and Hyperparameter Settings

Each training slide contained between 10 to 1000 tiles (median 372). During the autoencoder and PathCap training, we applied several data augmentation strategies similar to Liu et al. [2017b] to improve model robustness. First, we randomly applied left-right and top-down flips. Second, we perturbed color: brightness with a maximum delta of  $64/255$ , saturation with a maximum delta of 0.25, hue with a maximum delta of 0.04, and contrast with a maximum delta of 0.75. The Adam optimizer [Kingma and Ba, 2014] and validation data were used for parameter learning. Both the ResNet-18 for thumbnails and tiles were fine-tuned with a learning rate of  $1e-4$ . The decoder’s learning rate was  $4e-4$ . We decay the learning rate with factor 0.8 if there is no improvement for 8 consecutive epochs, and terminate training if there is no improvement for 20 consecutive epochs.

### 3.2.5 Cohort

We downloaded all clinical slides from the Genotype-Tissue Expression (GTEx) portal.<sup>1</sup> The GTEx project aims to provide the scientific community with a common resource with which to study human gene expression and regulation and its relationship to genetic variation. Notably, the GTEx Portal also provides open access to histopathology imaging data of donor tissue and histopathology notes describing the tissue sample quality. An example can

1. <http://gtexportal.org/home/histologyPage>

be seen in Figure. 3.5.

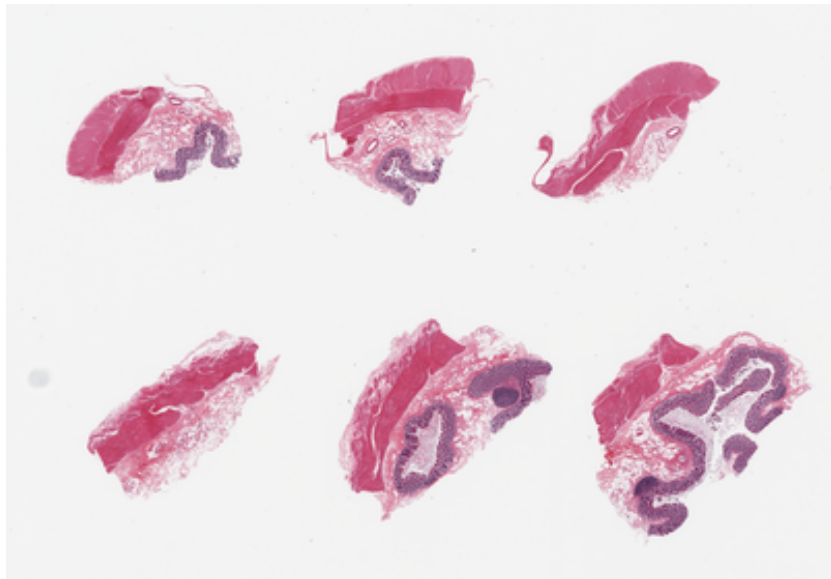


Figure 3.5: Example slide and caption from GTEx sample GTEx-131XE-0826: *6 pieces; 4 pieces have full thickness elements with well preserved mucosa; 2 have no mucosa (in this section).*

After selecting slides with captions and removing slides with sparse tissue content, we curated 9727 slide-caption pairs spanning 41 different tissue types. These pairs were randomly split into 7795 training, 948 validation, and 984 sized testing sets.

For the imaging data, we did not use any preprocessing methods on the whole-slide images. All histopathology slide images were subjected to digital tissue segmentation and segmented regions were clipped into non-overlapping 1000x1000px sized sections at 20x magnification. We removed tiles with an intensity greater than 0.70 or less than 0.2 to remove the background. For the caption data, all the captions were converted to lowercase. Tokens with less than 5 frequency were removed from the captions, resulting in 971 tokens that cover 95.06% word occurrences in the dataset.

### 3.3 Results on Real Data

#### 3.3.1 Results on Caption Generation

We first compared PathCap to a baseline model, which only takes low-resolution thumbnails as input and uses the Xu et al. [2015] approach in Table 3.1. For each step generating words, the model follows an attention mechanism and gives a weight for the spatial features extracted from thumbnails by ResNet-18 [He et al., 2016]. We used the Microsoft COCO [Chen et al., 2015a] tool to quantitatively compare the performance of models with different inputs. Here we used *beam size* = 1 and metrics including BLEU (columns labeled B-1, B-2, B-3, and B-4) [Papineni et al., 2002], Meteor [Denkowski and Lavie, 2014], Rouge-L [Lin, 2004] and CIDEr Vedantam et al. [2015]. We also examined a version of PathCap that only used tiles and without access to a thumbnail view, and found that using tiles alone performed slightly better than the baseline model. Taken together, PathCap, which combines information from high-resolution tile and low-resolution thumbnail views performed the best. All the metrics of PathCap are averaged over 20 rounds of testing.

Table 3.1: Performance on test set

Method	B-1	B-2	B-3	B-4	METEOR	ROUGE_L	CIDEr
Baseline	0.3822	0.2833	0.1996	0.1377	0.1958	0.4282	0.8936
<b>PathCap</b>	<b>0.4046</b>	<b>0.2986</b>	<b>0.2114</b>	<b>0.1455</b>	<b>0.2059</b>	0.4290	<b>0.9038</b>
Tiles-only	0.3944	0.2905	0.2040	0.1383	0.2032	<b>0.4312</b>	0.9003

#### 3.3.2 Results on Metric Learning

In order to demonstrate the superiority of triplet loss on tile embeddings, we trained two autoencoders. One autoencoder was trained only with reconstruction (mean squared error, MSE) loss. The other autoencoder was trained with reconstruction loss and triplet loss. The encoder part of the autoencoder was composed of two convolutional layers and two maxpooling layers. The output of the encoder (embedding) is of length 460. The decoder

part contained three convolutional layers. The  $\mu$  was set to 0.1, and the margin 0.001. We trained two separate PathCap models with the clusters using the representations from each of the two different autoencoders.

We observed that the two different autoencoders produced qualitatively different tile clusterings (Figure. 3.4). Next, we used the Microsoft COCO [Chen et al., 2015a] tool again to quantitatively compare the performance of models with different metrics, including BLEU [Papineni et al., 2002], Meteor [Denkowski and Lavie, 2014], Rouge-L [Lin, 2004] and CIDEr [Vedantam et al., 2015]. Table 3.2 shows the performance of our models when we used different metric learning methods for clustering. As above, B-1, B-2, etc. refer to the BLEU score. Overall, we demonstrate both a qualitative improvement in tile-level clustering and a quantitative improvement in caption generation using metric learning.

Table 3.2: Influence of triplet loss

Loss	B-1	B-2	B-3	B-4	METEOR	ROUGE_L	CIDEr
MSE only	0.3944	0.2878	0.2011	0.1381	0.2005	0.4219	0.8703
<b>MSE &amp; triplet loss</b>	<b>0.4046</b>	<b>0.2986</b>	<b>0.2114</b>	<b>0.1455</b>	<b>0.2059</b>	<b>0.4290</b>	<b>0.9038</b>

### 3.3.3 Results on Clustering

In order to explore the influence of cluster number  $K$ , we trained models with  $K$  from 2 to 5. An autoencoder was trained with reconstruction loss and triplet loss to generate embeddings for tiles extracted from each slide. After training the autoencoders, we generated representations for all tiles and performed k-means clustering using  $K$  from 2 to 5. In order to generate confidence intervals, we repeated this process 20 rounds.

For each PathCap trained model for each  $K$ , we evaluated our prediction on the testing dataset over 20 rounds. The average metrics over 20 rounds are reported in Table 3.3. The corresponding 95% confidence interval (CI) for each metric when cluster number = 3 are B-1 [0.3981,0.4111], B-2 [0.2938,0.3035], B-3 [0.2067,0.2162], B-4 [0.1406,0.1504], METEOR [0.2018,0.2100], ROUGE\_L [0.4232,0.4348] and CIDEr [0.8598,0.9478]. Overall, our analysis

suggests that PathCap is robust to cluster size changes, and demonstrates stable metrics across  $K$  from 2 to 5.

Table 3.3: Performance of PathCap with different cluster number ( $K$ )

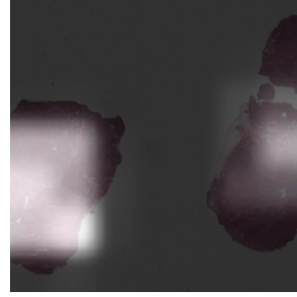
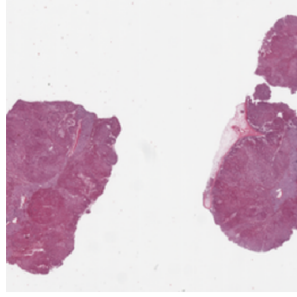
$K =$	B-1	B-2	B-3	B-4	METEOR	ROUGE_L	CIDEr
2	0.3797	0.2814	0.1976	0.1334	0.1973	0.4249	0.8627
3	<b>0.4046</b>	<b>0.2986</b>	<b>0.2114</b>	<b>0.1455</b>	<b>0.2059</b>	0.4290	0.9038
4	0.3887	0.2863	0.2003	0.1355	0.1990	0.4280	0.8989
5	0.3885	0.2909	0.2084	0.1447	0.2015	<b>0.4367</b>	<b>0.9621</b>

### 3.3.4 Results on Visualization

PathCap has the advantage of visualizing the caption prediction based on the attention weight given to tiles from a cluster. As a reference, visualization using the standard baseline model [Xu et al., 2015] is depicted in Figure. 3.6. The visualization and interpretation of the attention weight on the whole-slide images can be very ambiguous and non-specific.

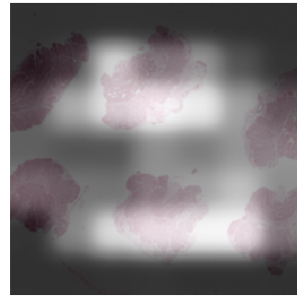
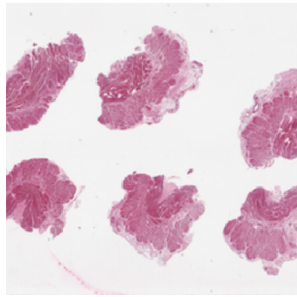
In contrast, with PathCap, an attention mechanism over tile features is deployed for our models. These tiles are sampled from different clusters. The clustering of tiles based on the embeddings learned using triplet loss underlies the potential of better separating the whole slides by small tiles. After the model is trained, weights on different clusters can be shown on the whole slide in the test dataset when the model predicts each word. We observe the model attends at word-level to both the inner parts of the tissue or texture and also the boundaries, depending on the caption context. Examples are shown in the Table 3.4.

Expert evaluation of the examples demonstrates broadly coherent and interpretable results. In the liver example, the predicted caption and visualization are appropriate for macrovesicular steatosis. Next, for the Esophagus example, the use of the phrase “good specimens” in the prediction is highly subjective and likely an atypical way to annotate specimens. However, the detection of muscularis propria provides improved context relative to the reference caption. For the skin example, the prediction of “5% dermal fat” is appropri-



(a) Thumbnail for GTEX-15CHR-0625

(b) Attention weight for word "myometrium"



(c) Thumbnail for GTEX-Y3I4-0925

(d) Attention weight for word "muscularis"

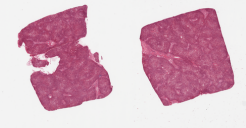
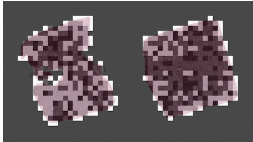
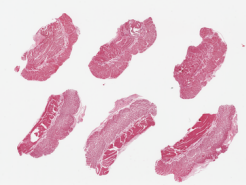
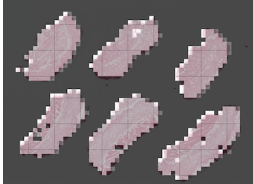




Figure 3.6: Example of visualizing caption tokens with a standard baseline model [Xu et al., 2015]. (a) and (c) are the input thumbnails to the model. (b) and (d) show the attention weights when the model generates the "myometrium" and "muscularis" tokens respectively. White/bright indicates more attention weight, black/dark indicates less attention weight.

ate, however, the tile clusters visualized for the "fat" token are instead squamous epithelium. Finally, for the colon example, the predicted caption and visualization are correct in that the full thickness section contains about 1 mm thickness of colon, but it is mostly an irrelevant measure. Notably, the caption neglected to capture autolytic properties from the autopsy material.

### 3.4 Discussion

In this work, we present and examine the complex task of generating short, clinically relevant captions from gigapixel whole-slide histopathology images. We show that clustering

Table 3.4: Visualization of the PathCap method on four test slides from four different tissues. The last column shows some examples of attention weights when the model generates the corresponding tokens. White/bright indicates more attention weight, and black/dark indicates less attention weight.

Slide	PathCap Prediction	Reference	Example
 <p>Liver<sup>a</sup></p> <p><sup>a</sup>. GTEx sample ID: 13FLV-0326</p>	<p>2 pieces, diffuse macrovesicular steatosis involves 70 % of parenchyma</p>	<p>2 pieces; includes a portion of the capsule ( target is 1 cm below capsule ), mild steatosis, passive congestion, focal portal chronic inflammation</p>	 <p>"macrovesicular"</p>
 <p>Esophagus<sup>b</sup></p> <p><sup>b</sup>. GTEx sample ID: 13FTW-1926</p>	<p>6 pieces , up to &lt;unk&gt; ; all muscularis , good specimens</p>	<p>6 pieces ; well trimmed</p>	 <p>"muscularis"</p>
 <p>Skin<sup>c</sup></p> <p><sup>c</sup>. GTEx sample ID: 13NYS-0126</p>	<p>6 pieces ; well trimmed ; 5 % dermal fat</p>	<p>6 pieces ; &lt;unk&gt; epidermis ( &lt;unk&gt; ) , solar elastosis ; well trimmed , 10 % dermal fat</p>	 <p>"fat"</p>
 <p>Colon<sup>d</sup></p> <p><sup>d</sup>. GTEx sample ID: 13O3P-2326</p>	<p>6 pieces , mucosa up to 1mm , &lt;unk&gt; % thickness</p>	<p>6 pieces ; mucosa autolyzed ; muscularis preserved</p>	 <p>"mucosa"</p>



tiles based on the embeddings learned using triplet loss allows for coherent segmentation of whole-slide images and results in improved visualization of attention. Thus, our specific technical contribution of clustering tiles within histopathology images in order to facilitate downstream tasks, such as caption generation and interpretation, suggests a promising strategy for other machine learning tasks in digital pathology. Finally, we demonstrate the relative effectiveness of PathCap compared to a standard baseline caption prediction approach and propose the GTEEx dataset as a novel benchmark for future caption prediction and interpretation methods.

**Limitations** We note some important limitations in our work. First, while PathCap achieves better performance over the standard baseline caption prediction method, there is significant room for improvement. Our results confirm that caption generation from histopathology images is a unique and technically challenging problem. Future work in caption prediction could benefit from considering this specific problem setting. Second, we trained and tested our model only on the GTEEx data. Due to limitations in publicly available paired caption and histology images, we were unable to evaluate domain adaptation or consider other imaging datasets. Future work should consider evaluating PathCap generated captions on additional datasets as they become available. Third, our captions are short descriptions relating to specimen quality from GTEEx (e.g., sample composition). We did not test our model on text from large reports or clinical notes. We hypothesize that integration of a hierarchical LSTM model, such as one proposed by Jing et al. [2017], may be useful for these scenarios.

# CHAPTER 4

## HYPERBOLIC ATTENTION MODEL FOR HISTOPATHOLOGY IMAGES

### 4.1 Motivation

In the field of computer vision, deep learning methods have made significant strides over the past decade, achieving remarkable performance in image classification and image retrieval tasks. Deep neural networks have demonstrated their ability to learn intricate features directly from various biomedical imaging modalities, including X-ray, CT, MRI, and histopathology images, and have been successfully applied to complex tasks like disease prediction and outcome analysis [Litjens et al., 2017, Shen et al., 2017]. These networks typically employ a sequence of convolutional transformations to encode images into embeddings within Euclidean space. However, emerging evidence suggests that certain types of data, particularly those exhibiting hierarchical or multi-scale structures, may not be efficiently represented in Euclidean space [Nickel and Kiela, 2017, Ganea et al., 2018a, Gulcehre et al., 2018, Chami et al., 2019]. Therefore, there is a growing interest in exploring alternative spaces, such as hyperbolic geometry, to more effectively capture and model the intrinsic properties of such data.

Simultaneously, medical images, such as histopathology images, often pose challenges due to their large size, surpassing the capacity of most modern GPUs and off-the-shelf deep learning models. Consequently, various approaches have been adopted to handle this issue, including image rescaling to lower resolutions or segmenting images into smaller tiles that fit into GPU memory. However, these solutions present trade-offs. Rescaling images can lead to distortion and the loss of crucial image details, while image segmentation into tiles, although high-resolution, may sacrifice contextual and spatial information. In this context, a compelling question arises: Is there a middle ground where we can integrate differently

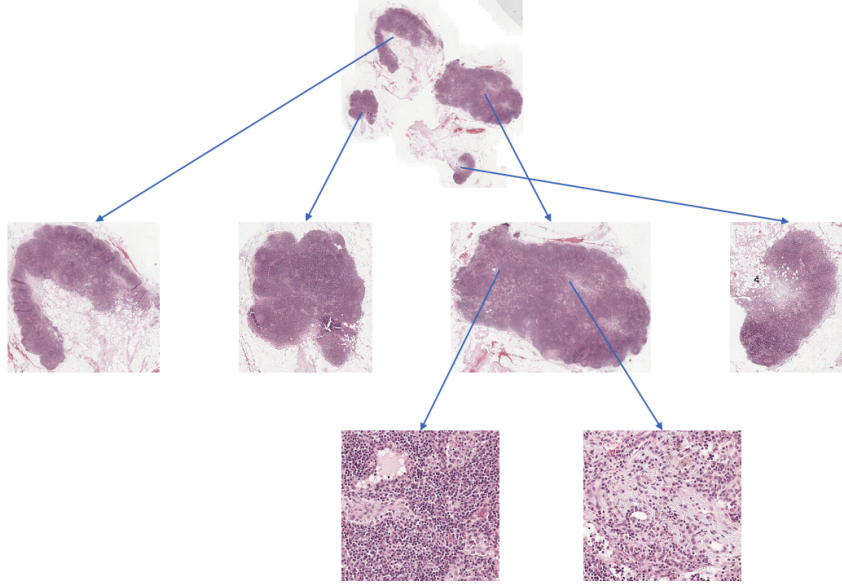


Figure 4.1: Example digital hematoxylin and eosin (H&E) stained histopathology slide image with differently scaled views and the relative hierarchy.

scaled versions of an image to harness both lower-resolution contextual information and higher-resolution detailed views?

Histopathology images exhibit a hierarchical or multi-scale structure, as they can be examined at various magnifications or scales. Surprisingly, the exploration of robust approaches to effectively utilize multi-scaled views in histopathology images remains limited. In this study, we introduce and evaluate a straightforward hyperbolic modification to existing deep learning architectures, enabling the integration of information from different scales Figure. 4.1.

Hyperbolic space, although diffeomorphic to standard Euclidean space, possesses a constant negative sectional curvature. The Poincaré ball model serves as a valuable representation of hyperbolic space, wherein the distance from the origin to the boundary exponentially increases. This property renders Poincaré embeddings akin to continuous analogs of trees [Nickel and Kiela, 2017]. Consequently, they offer a suitable framework for learning embeddings that capture natural hierarchies, such as different magnifications of image sections [Ganea et al., 2018a]. Our approach draws inspiration from recent advancements that lever-

age Poincaré embeddings to model hierarchical structures [Gulcehre et al., 2018, Chami et al., 2019, Khrulkov et al., 2020].

We propose a hyperbolic counterpart to the standard attention-based models commonly utilized in medical imaging. Through empirical examples, we demonstrate that leveraging hyperbolic spaces can potentially enhance the performance of tissue classification tasks in H&E images. Our contributions encompass the following: (1) an extension of previous research in computer vision [Khrulkov et al., 2020] by introducing a novel formulation of hyperbolic attention; (2) compelling evidence showcasing the potential benefits of modeling multi-scale views of H&E images using hyperbolic spaces. Finally, we conclude the paper by discussing current challenges and illuminating future opportunities. Parts of this chapter are replicated from Zhang et al. [2020a] with some modifications.

## 4.2 Method

We introduce a hyperbolic generalization to traditional CNN architectures in order to extract visual features from different scales and perform operations in hyperbolic space. Briefly, a traditional CNN, such as ResNet18 He et al. [2016], can be used to generate feature representations of tiles extracted from whole-slide H&E images. Next, we define a bijective mapping of the generated features to hyperbolic space. We then define linear, multinomial regression and attention layers operating in hyperbolic space. We note the attention layer operates in hyperbolic space and computes a slide-level representation. Finally, we define classification using a multinomial regression layer. We derive and present our formal definitions in the following subsections and adopt notations from Ganea et al. [2018b] and Khrulkov et al. [2019]. We denote the resulting model as a hyperbolic-attention model.

### 4.2.1 Poincaré Ball Model

The Poincaré model  $(\mathbb{D}^n, g^{\mathbb{D}})$  is defined by the manifold  $\mathbb{D}^n = \{x \in \mathbb{R}^n : \|x\| < 1\}$  equipped with Riemannian metric  $g_x^{\mathbb{D}} = \lambda_x^2 g^E$  where  $\lambda_x := \frac{2}{1-\|x\|^2}$ .  $g^E = \mathbf{I}_n$  is the Euclidean metric tensor. To make use of the Poincaré ball of radius  $c \geq 0$ , we denote  $\mathbb{D}_c^n := \{x \in \mathbb{R}^n : |c\|x\|^2 < 1\}$ . If  $c = 0$ ,  $\mathbb{D}_c^n = \mathbb{R}^n$ ; If  $c > 0$ , it is a open ball with radius  $1/\sqrt{c}$ .

### 4.2.2 Möbius Addition

For a pair of  $\mathbf{x}, \mathbf{y} \in \mathbb{D}_c^n$ , the Möbius addition is defined as follows

$$\mathbf{x} \oplus_c \mathbf{y} := \frac{(1 + 2c\langle \mathbf{x}, \mathbf{y} \rangle + c\|\mathbf{y}\|^2) \mathbf{x} + (1 - c\|\mathbf{x}\|^2) \mathbf{y}}{1 + 2c\langle \mathbf{x}, \mathbf{y} \rangle + c^2\|\mathbf{x}\|^2\|\mathbf{y}\|^2} \quad (4.1)$$

With  $c \rightarrow 0$  we can obtain the Euclidean distance of two vectors in  $\mathbb{R}^n$ .

### 4.2.3 Exponential and Logarithmic Maps

In order to do operations in hyperbolic space, bijective maps are defined to map from  $\mathbb{R}^n$  to  $\mathbb{D}_c^n$ . The exponential map  $\exp_x^c$  is a function from  $\mathbb{R}^n$  to  $\mathbb{D}_c^n$

$$\exp_{\mathbf{x}}^c(\mathbf{v}) := \mathbf{x} \oplus_c \left( \tanh \left( \sqrt{c} \frac{\lambda_{\mathbf{x}}^c \|\mathbf{v}\|}{2} \right) \frac{\mathbf{v}}{\sqrt{c}\|\mathbf{v}\|} \right) \quad (4.2)$$

The inverse map is defined as

$$\log_{\mathbf{x}}^c(\mathbf{y}) := \frac{2}{\sqrt{c}\lambda_{\mathbf{x}}^c} \operatorname{arctanh} \left( \sqrt{c} \|\mathbf{x} \oplus_c \mathbf{y}\| \right) \frac{-\mathbf{x} \oplus_c \mathbf{y}}{\|\mathbf{x} \oplus_c \mathbf{y}\|} \quad (4.3)$$

In practice, we use the maps  $\exp_0^c$  and  $\log_0^c$  for transition between the Euclidean and Poincaré ball representations of a vector.

#### 4.2.4 Hyperbolic Linear Layer

Similar to Khrukov et al. [2019], we define a hyperbolic linear layer a map from  $\mathbb{D}_c^{n_1}$  to  $\mathbb{D}_c^{n_2}$ . For input  $\mathbf{x} \in \mathbb{D}_c^{n_1}$  to this layer and a trainable matrix  $\mathbf{M}$  of size  $n_2 \times n_1$ , if  $\mathbf{M}\mathbf{x} \neq 0$ , the output of this layer is

$$M^c(\mathbf{x}) := \frac{1}{\sqrt{c}} \tanh \left( \frac{\|\mathbf{M}\mathbf{x}\|}{\|\mathbf{x}\|} \operatorname{arctanh}(\sqrt{c}\|\mathbf{x}\|) \right) \frac{\mathbf{M}\mathbf{x}}{\|\mathbf{M}\mathbf{x}\|} \quad (4.4)$$

otherwise  $M^c(\mathbf{x}) := 0$ . For a bias vector  $\mathbf{b} \in \mathbb{D}_c^{n_2}$ , the corresponding linear layer is  $M^c(\mathbf{x}) \oplus_c \mathbf{b}$ .

#### 4.2.5 Klein Model

In order to define the hyperbolic attention model, we will make use of the Klein model and hyperbolic averaging. Similar to Poincaré model, it is defined in  $\mathbb{K}^n = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\| < 1\}$ . Let  $\mathbf{x}^{\mathbb{D}}$  and  $\mathbf{x}^{\mathbb{K}}$  denote the coordinates of the same point in the Poincaré and Klein models. We use the following formulas to map from each other.

$$\mathbf{x}^{\mathbb{D}} = \frac{\mathbf{x}^{\mathbb{K}}}{1 + \sqrt{1 - c \|\mathbf{x}^{\mathbb{K}}\|^2}} \quad (4.5)$$

$$\mathbf{x}^{\mathbb{K}} = \frac{2\mathbf{x}^{\mathbb{D}}}{1 + c \|\mathbf{x}^{\mathbb{D}}\|^2} \quad (4.6)$$

#### 4.2.6 Hyperbolic Attention

Given a set of  $\mathbf{v}_i^{\mathbb{D}} \in \mathbb{D}^n$ , the corresponding coordinates in Klein model are  $\mathbf{v}_i^{\mathbb{K}}$ , we define the attention weights  $\alpha_i$  as follows.

$$\alpha_i = f \left( \mathbf{v}_i^{\mathbb{D}} \right) \quad (4.7)$$

The function  $f(\cdot)$  is a hyperbolic neural network followed by softmax or sigmoid. The outputs  $m^{\mathbb{K}}$  for the hyperbolic attention module are as follows.

$$m^{\mathbb{K}}\left(\{\alpha_i\}, \{\mathbf{v}_i^{\mathbb{K}}\}\right) = \sum_i \frac{\alpha_i \gamma\left(\mathbf{v}_i^{\mathbb{K}}\right) \mathbf{v}_i^{\mathbb{K}}}{\sum_\ell \alpha_\ell \gamma\left(\mathbf{v}_\ell^{\mathbb{K}}\right)} \quad (4.8)$$

where the  $\gamma\left(\mathbf{v}_i^{\mathbb{K}}\right)$  are the Lorentz factors,

$$\gamma\left(\mathbf{v}_i^{\mathbb{K}}\right) = \frac{1}{\sqrt{1 - c \|\mathbf{v}_i^{\mathbb{K}}\|^2}} \quad (4.9)$$

After we get the hyperbolic attention output  $m_i^{\mathbb{D}}$  of the Klein model, we can map it to the Poincaré model.

#### 4.2.7 Multiclass Logistic Regression

The resulting formula for hyperbolic multiclass logistic regression for  $K$  classes is written below; here  $p_k \in \mathbb{D}_c^n$  and  $a_k \in T_{\mathbf{p}_k} \mathbb{D}_c^n \setminus \{\mathbf{0}\}$  are learnable parameters.

$$p(y = k | \mathbf{x}) \propto \exp\left(\frac{\lambda_{\mathbf{p}_k}^c \|\mathbf{a}_k\|}{\sqrt{c}} \operatorname{arcsinh}\left(\frac{2\sqrt{c} \langle -\mathbf{p}_k \oplus_c \mathbf{x}, \mathbf{a}_k \rangle}{(1 - c \|\mathbf{p}_k \oplus_c \mathbf{x}\|^2) \|\mathbf{a}_k\|}\right)\right) \quad (4.10)$$

### 4.3 Results

We compare the performance of our hyperbolic-attention model with a baseline model and the *Deep MIL* attention-based model on simple tissue classification tasks using two well-known public datasets. We provide a thorough description of each experiment in the following subsections.

### 4.3.1 *Camelyon16*

The Camelyon16 challenge Litjens et al. [2018] was organized by the IEEE International Symposium on Biomedical Imaging. It evaluated various machine learning models to detect cancer metastasis. There are 159 slides with normal tissue class labels and 111 slides with tumor tissue class labels in the training set. The test data set contains 80 slides with normal tissue and 50 slides with tumor tissue. In this work, we evaluate our model on classifications based on slide-level annotations. We perform minimal data pre-processing. Tile sizes of 500x500 pixels (500px) and 1000x1000 pixels (1000px) are extracted without overlap, and 2000x2000 pixels (2000px) tiles are extracted with step size 1000. In order to filter out non-tissue containing or background tiles, we only keep those tiles with average intensity less than 0.85 and greater than 0.2.

We implemented a baseline approach similar to Coudray et al. [2018]. We used ResNet18 He et al. [2016] pretrained on ImageNet Russakovsky et al. [2015a]. We fine-tuned the model by updating all layers to classify tiles extracted from H&E slides. Tiles of different sizes were re-scaled to the default ResNet18 input layer size and generated embeddings of length 10. We labeled all tiles with the same label as the slide from which they were extracted. During validation or testing, the model aggregated all the tile predictions of a slide by taking the average and using this average as a slide-level prediction.

We implemented the *Deep MIL* attention-model by randomly sampling 5 tiles as input for each slide. We used ResNet18 pretrained on ImageNet as a feature extractor. We again fine-tuned the model by updating all layers to classify tiles extracted from H&E slides. Similarly, tiles of different sizes were re-scaled to the default ResNet18 input layer size and generated embeddings of length 10. We performed test-time augmentation for each slide in the test data set, where the model generated a prediction for each test slide 10 times and randomly sampled 5 tiles each time. We calculated the average of all the 10 predictions as the final output.



We sought to compare performance between the baseline model and the *Deep MIL* attention-based model using tiles from a single fixed scale view. We independently evaluated performance on 3 different scales (Table I). The models were tested in a 5-fold cross-validation manner. We choose 4 folds as training and 1 fold as a validation data set for each assignment. We trained the models 4 times for each assignment. Both models were trained with the Adam optimization method with learning rate=1e-4,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ , and  $\epsilon=1e-8$ . To ensure numerical stability, clipping by norm, similar to Khrulkov et al. [2019], is performed. For all the 20 checkpoints of each model, we tested the performance on the test dataset and the mean AUC and confidence interval (CI) are reported in Table 4.1. We can see that for the baseline model, the best tile size is 1000. *Deep MIL* performs similarly for the 3 scales and demonstrates it is always better than the baseline model.

Table 4.1: Performance of single scale models

Scale	Baseline model		Deep MIL	
	Mean AUC	CI(0.95)	Mean AUC	CI(0.95)
500px	0.569	[0.474,0.663]	0.619	[0.524,0.714]
1000px	0.597	[0.467,0.726]	0.602	[0.505,0.700]
2000px	0.583	[0.460,0.706]	0.613	[0.540,0.687]

Next, we sought to compare the performance between the *Deep MIL* attention-model and our hyperbolic attention model in a multi-scale setting. We combined and used 15 tiles, 5 tiles from each of the previous 3 scales, to train and evaluate slide-level classification accuracy. We also examined the effect of different embedding lengths, by setting lengths to 5, 10, and 100. The results are reported in Table 4.2. By combining tiles from different scales, the *Deep MIL* model does not outperform a single scale. However, the hyperbolic attention model is better than *Deep MIL* when using multi-scale views. This suggests that our hyperbolic attention model can learn better multi-scale embeddings.

Table 4.2: Performance of multiple scale models

Embed size	Deep MIL		Hyperbolic attention	
	Mean AUC	CI(0.95)	Mean AUC	CI(0.95)
5	0.602	[0.513,0.690]	0.623	[0.536,0.710]
10	0.606	[0.519,0.693]	0.615	[0.537,0.692]
100	0.592	[0.491,0.694]	0.637	[0.574,0.700]

### 4.3.2 TCGA

We consider a second general tissue classification task involving normal and lung cancer subtypes (LUAD and LUSC) as presented in Coudray et al. [2018]. We downloaded H&E lung slides from the TCGA Genomic Data Commons Grossman et al. [2016]. There were 811 LUAD slides, 745 LUSC slides, and 585 adjacent normal slides. We processed all tiles in the same way as we did for the Camelyon16 data set.

We evaluated our model with the baseline model and the *Deep MIL* model on the lung classification task. All slides were again split into 5-fold and we tested all models on the lung data set in a 5-fold cross-validation manner. The models were trained 4 times with the same settings as with the Camelyon16 dataset. The mean Macro-average AUC and CI(0.95) of 20 checkpoints are reported in Table 4.3 and Table 4.4.

Table III shows the performance of baseline models and *Deep MIL* models trained on 3 different scales. We find that the *Deep MIL* models produce slightly better results.

Table 4.3: Performance of single scale models

Scale	Baseline model		Deep MIL	
	Mean AUC	CI(0.95)	Mean AUC	CI(0.95)
500px	0.969	[0.954,0.985]	0.973	[0.957,0.989]
1000px	0.971	[0.955,0.986]	0.972	[0.954,0.990]
2000px	0.966	[0.945,0.987]	0.967	[0.949,0.985]

Table 4.4 shows the performance of *Deep MIL* and our hyperbolic attention model when

we combine tiles from 3 scales. Note that in this example, single-scale and multiple-scale *Deep MIL* produce comparable results. Our model’s performance is better when the embedding size is 5 or 10. When the embedding size is 100, the performances of the two models are comparable. Overall, this again suggests that our hyperbolic attention model can learn better multi-scale embeddings.

Table 4.4: Performance of multiple scale models

<b>Embed size</b>	<b>Deep MIL</b>		<b>Hyperbolic attention</b>	
	<b>Mean AUC</b>	<b>CI(0.95)</b>	<b>Mean AUC</b>	<b>CI(0.95)</b>
5	0.967	[0.952,0.983]	0.971	[0.958,0.985]
10	0.970	[0.948,0.992]	0.974	[0.961,0.988]
100	0.973	[0.958,0.988]	0.972	[0.958,0.986]

## 4.4 Discussion and conclusion

In this paper, we develop a hyperbolic-attention model using a Poincaré ball and Klein model to classify histopathology slide images. Our results suggest that our hyperbolic attention model can efficiently learn multi-scale embeddings. However, we note some important limitations in our work and plan to explore these in future work. First, while our hyperbolic-attention model achieves better performance over the standard baseline and the *Deep MIL* model, there is still room for improvement, especially when the slide number is limited. Second, we did not explore an integrative interpretation of hyperbolic space geometry and the use of attention to identify salient structures and scales that are associated with improved model performance. Third, while we evaluated our model on two independent datasets, we think future work should also examine additional datasets and data types, such as X-ray images.

# CHAPTER 5

## ENHANCING INSTANCE-LEVEL IMAGE CLASSIFICATION WITH SET-LEVEL LABELS

### 5.1 Motivation

A large amount of labeled data from the source domain 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. In the source domain, though, 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 and 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., 2020b, 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.

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. 5.1a) 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. 5.1a) are easily available. This motivates us to leverage these abundant and cost-efficient coarse-grained labels and hierarchical relationships, such as between organs and tissues (as depicted in Figure. 5.1b), 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. 5.1a. We emulate a programmatic labeler that uses heuristics such as keywords, regular expressions, 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.

In this chapter, we introduce Fine-grAined representation learning from Coarse-grAined LabEls (FACILE), a novel generic representation learning framework that uses easily accessible coarse-grained annotations to quickly adapt to new fine-grained tasks. Distinct from

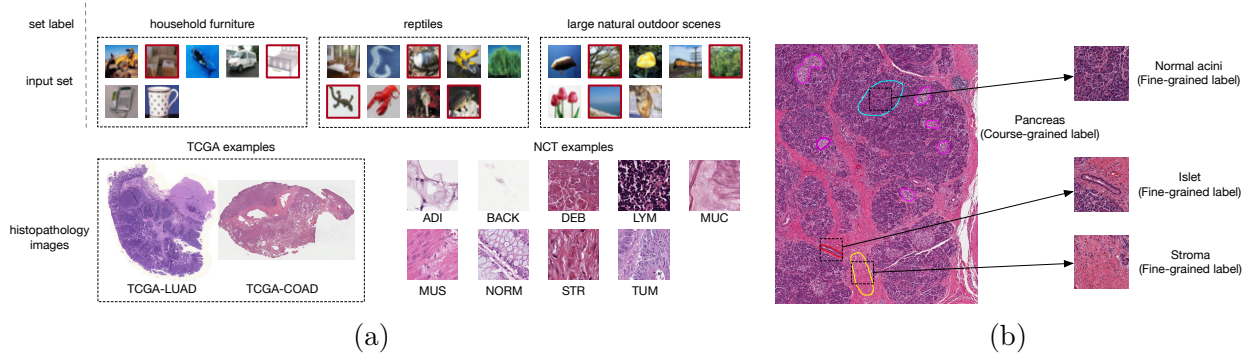


Figure 5.1: (a) A collection of image sets sampled from CIFAR-100 are in the upper row. The coarse-grained label of a set is the most frequent superclass of images inside the set. WSI examples from TCGA and patches from NCT dataset are in the lower row. (b) Hierarchy of coarse- and fine-grained labels for histopathology images.

existing practices in FSL and TL, 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. Parts of this chapter are replicated from Zhang et al. [2023a] with some modifications.

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 excess risk 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. The generated datasets are used to evaluate different models. We also evaluate models by fine-

tuning the fully connected layer appended to ViT-B/16 [Dosovitskiy et al., 2020] of CLIP [Radford et al., 2021] in an anomaly detection dataset based on CUB200 [He and Peng, 2019]. 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 13% with logistic regression fine-grained classifier. We further evaluate various models by fine-tuning the fully connected layer appended to ViT-B/14 [Dosovitskiy et al., 2020] of DINO V2 [Oquab et al., 2023]. These models can leverage the capability of “foundation” models and enhance the model performance on target tasks. Our experiments provide compelling evidence of the efficacy and generalizability of FACILE across various datasets, highlighting its potential as a robust representation learning framework.

## 5.2 Fine-Grained Representation Learning from Coarse-Grained Labels

**Notations** Our model pretrains on a collection of samples, denoted by  $\{(s_i, w_i)\}_{i=1}^m$ . Each  $s_i$  is a set of instances  $\{x_j\}_{j=1}^a$ , where  $a$  is the set size that can vary.  $\{w_i\}$  are the 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  $\mathcal{S} = \{\{x_1, \dots, x_a\} : x_j \in \mathcal{X} \text{ for } \forall j \in [a]\}$ . 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  $x \in \mathcal{X}$ . The model could benefit from the information on the coarse-grained labels.

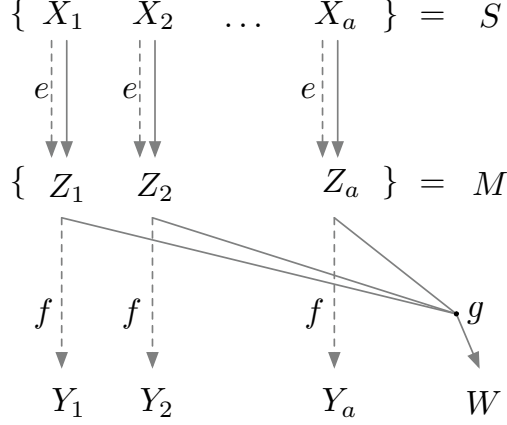


Figure 5.2: Schema of the FACILE model. The dotted lines represent the flow of fine-grained data, and the solid lines denote the flow of coarse-grained labels.

### 5.2.1 The FACILE Algorithm

We study the model in a 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  $\mathcal{E} = \{e : \mathcal{X} \rightarrow \mathcal{Z}\}$ , set-input functions  $\mathcal{G} = \{g : \mathcal{M} \rightarrow \mathcal{W}\}$  where  $\mathcal{M} = \{\{z_1, \dots, z_a\} : z_j \in \mathcal{Z} \text{ for } j \in [a]\}$ , and fine-grained label predictors  $\mathcal{F} = \{f : \mathcal{Z} \rightarrow \mathcal{Y}\}$ . The corresponding set-input feature map of an instance feature map  $e$  is defined as  $\phi^e : \mathcal{S} \rightarrow \mathcal{M}$ . 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. 5.2.

We assume two loss functions:  $\ell^{\text{fg}} : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$  for fine-grained label prediction and  $\ell^{\text{cg}} : \mathcal{W} \times \mathcal{W} \rightarrow \mathbb{R}$  for coarse-grained label prediction.  $\ell^{\text{fg}}$  measures the loss of the fine-grained label predictor. We assume this loss is differentiable in its first argument.  $\ell^{\text{cg}}$  measures the loss of pretraining with coarse-grained labels. For theoretical analysis, we are interested in two particular cases of  $\ell^{\text{cg}}$ : i)  $\ell^{\text{cg}}(w, w') = \mathbb{1}\{w \neq w'\}$  where  $\mathcal{W}$  is a categorical



---

**Algorithm 1** FACILE algorithm
 

---

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

space; and ii)  $\ell^{\text{cg}}(w, w') = \|w - w'\|$  (for some norm  $\|\cdot\|$  on  $\mathcal{W}$ ) where  $\mathcal{W}$  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:

$$\mathbb{E}_{P_{X,Y}}[\ell^{\text{fg}}(\hat{f} \circ \hat{e}(X), Y) - \ell^{\text{fg}}(f^* \circ e^*(X), Y)] \quad (5.1)$$

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. 5.3. Given an input set  $s_i$  comprising instances  $x_1, \dots, x_a$ , the feature map  $e$  is 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 framework is designed to be a generic algorithm that is compatible with any supervised learning method in its pretraining stage. We chose SupCon (Supervised Contrastive Learning) [Khosla et al., 2020] and FSP as they are representative of the two main approaches within supervised learning: contrastive and non-contrastive (traditional supervised) learning, respectively. During testing, we extract the pretrained feature map  $\hat{e}$  and fine-tune a classifier  $f$  using the generated embeddings

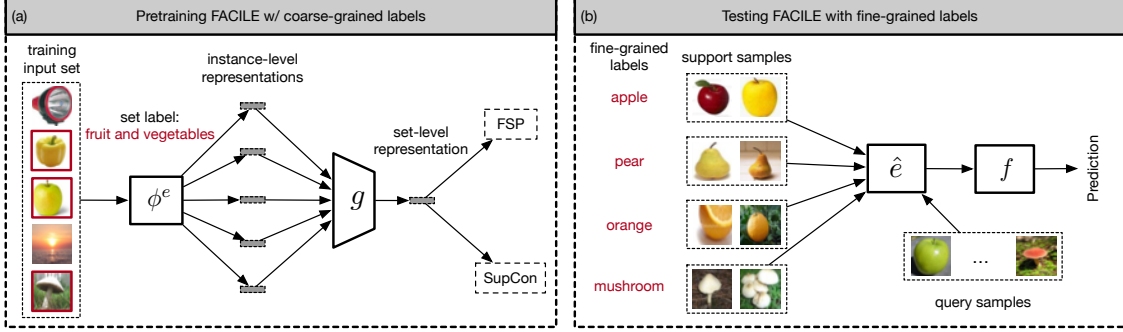


Figure 5.3: An overview of the FACILE algorithm. (a) Pretraining step of FACILE with coarse-grained labels. The input is a set of images and the target is set-level coarse-grained label.  $e$  is an instance feature map and  $\phi^e$  is the corresponding set-input feature map.  $g$  is the set-input model. We can instantiate the  $\mathcal{A}(\ell^{\text{cg}}, \mathcal{D}_m^{\text{cg}}, \mathcal{E})$  with any supervised learning algorithms, e.g., fully supervised pretraining (FSP) with cross-entropy loss and the SupCon model. (b) Fine-grained learning of FACILE with fine-grained labels. The learned instance feature map  $\hat{e}$  extracts instance-level features from patches of the support set and query set.  $f$  is the fine-grained label predictor.

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.

### 5.2.2 Theoretical Analysis

We denote the underlying distribution of  $\mathcal{D}_m^{\text{cg}}$  as  $P_{S,W}$  and the underlying distribution of  $\mathcal{D}_n^{\text{fg}}$  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  $\mathcal{D}_n^{\text{fg, aug}}$  is i.i.d. samples from  $\hat{P}_{Z,Y}$ . In order to include the underlying distribution of  $\mathcal{D}_m^{\text{cg}}$ , and  $\mathcal{D}_n^{\text{fg}}$  into analysis, with a slight abuse of notation we use  $\mathcal{A}_m(\ell^{\text{cg}}, P_{S,W}, \mathcal{E})$  to denote  $\mathcal{A}(\ell^{\text{cg}}, \mathcal{D}_m^{\text{cg}}, \mathcal{E})$  and use  $\mathcal{A}_n(\ell^{\text{fg}}, \hat{P}_{Z,Y}, \mathcal{F})$  to denote  $\mathcal{A}(\ell^{\text{fg}}, \mathcal{D}_n^{\text{fg, aug}}, \mathcal{F})$ . The two learning algorithms are described as follows.

**Definition 1.** (Coarse-grained learning; pretraining) Let  $\text{Rate}_m(\ell^{\text{cg}}, P_{S,W}, \mathcal{E}; \delta)$  (abbreviated to  $\text{Rate}_m(\ell^{\text{cg}}, P_{S,W}, \mathcal{E})$ ) be the rate of  $\mathcal{A}_m(\ell^{\text{cg}}, P_{S,W}, \mathcal{E})$  which takes  $\ell^{\text{cg}}$ ,  $\mathcal{E}$  and  $m$  i.i.d.

observations from  $P_{S,W}$  as input, and return a feature map  $\hat{e} \in \mathcal{E}$  such that

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

with probability at least  $1 - \delta$ .

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

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  $e, e' \in \mathcal{E}$ ,

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

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  $\mathcal{E}$ ,  $\ell^{\text{fg}}$  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  $\text{Rate}_m(\ell^{\text{cg}}, P_{S,W}, \mathcal{E}) = \mathcal{O}(1/m^\alpha)$ . Then when  $\mathcal{A}_n(\ell^{\text{fg}}, \hat{P}_{Z,Y}, \mathcal{F})$  is ERM we obtain excess risk  $\mathbb{E}_{P_{X,Y}} \left[ \ell_{\hat{f} \circ \hat{e}}^{\text{fg}}(X, Y) - \ell_{f^* \circ e^*}^{\text{fg}}(X, Y) \right]$  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  $\text{Rate}_m(\ell^{\text{cg}}, P_{S,W}, \mathcal{E}) = \mathcal{O}(1/\sqrt{m})$ , 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.,  $\text{Rate}_m(\ell^{\text{cg}}, P_{S,W}, \mathcal{E}) = \mathcal{O}(1/m)$ , one can obtain 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  $2L\text{Rate}_m(\ell^{\text{cg}}, P_{S,W}, \mathcal{E}) + \text{Rate}_n(\ell^{\text{fg}}, \hat{P}_{Z,Y}, \mathcal{F})$  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,  $\text{Rate}_n(\ell^{\text{fg}}, \hat{P}_{Z,Y}, \mathcal{F})$  is also bounded by Proposition 8. We refer interested readers to §A.8 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 §5.3.3 and §5.3.5. We also demonstrate the exceptional efficacy of the proposed algorithm compared to baseline models on natural image datasets and histopathology image datasets.

## 5.3 Results

### 5.3.1 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., 2020a, Caron et al., 2020, Grill et al., 2020, Chen and He, 2021] and

weakly supervised models [Donahue et al., 2014, Sun et al., 2017, Zeiler and 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., 2020a] for contrastive learning and SimSiam [Chen and He, 2021] for non-contrastive learning. Details of these self-supervised learning algorithms can be found in §A.7.

**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.

Following previous work in FSL [Tian et al., 2020b, 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).

### 5.3.2 *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. In our downstream tasks, we perform few-shot classifications of fine-grained classes. Despite being distinct from the downstream fine-grained labels, the coarse-grained labels offer useful information for learning useful representations for downstream tasks.

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

Table 5.1: Pretraining on input sets from CIFAR-100. Testing with 5-shot 5-way meta-test sets; average F1 and CI are reported.

The ResNet18 [He et al., 2016] is used as feature maps  $\hat{e}$ . For FACILE-FSP, we pretrain the feature map  $\hat{e}$  from these input sets and targets with  $\ell_1$  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 §A.1.1. 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 5.1. We can see that FACILE-FSP outperforms self-supervised learning models by a large margin.

### 5.3.3 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 §5.3.2 and can be found in §A.1.1. The performances of all models are reported in Table 5.1. We

can see that FACILE-FSP obtains better results compared to other models.

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  $\gamma$ . 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 the 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. 5.4. The figure shows the log-linear relationship of FACILE-FSP’s generalization error on downstream tasks w.r.t. fine-grained label number. This visualization effectively captures how coarse-grained label number  $m$  impacts the model’s generalization capabilities.

### 5.3.4 *Fine-tune CLIP Model with Anomaly Detection Dataset*

In this experiment, we sought to enhance model performance with coarse-grained labels of the anomaly detection datasets [Zaheer et al., 2017, Lee et al., 2019a]. A total of 11,788 input sets of size 10 are constructed from the CUB200 [He and Peng, 2019] training dataset by including one example that lacks an attribute common to the other examples in the input set. The coarse-grained labels are the positions of the anomalies. This setup creates a challenging scenario for models, as they must identify the outlier among otherwise similar instances. In downstream tasks, we evaluate the fine-tuned feature encoder composed of the fixed CLIP [Radford et al., 2021] image encoder ViT-B/16 and appended a fully connected layer on the classification of species of the CUB200 test dataset. The batch normalization [Ioffe and Szegedy, 2015] and ReLU are applied to the fully-connected layer.

Following this experiment setup, the rationale behind utilizing coarse-grained labels is

grounded in their potential to enhance model discernment in downstream tasks. By training the model to identify anomalies in sets where one item diverges from the rest, we essentially teach it to focus on subtle differences and critical attribute features. This enhanced focus is particularly beneficial for fine-grained classification tasks in the CUB200 test dataset, where distinguishing between closely related species requires the model to recognize and prioritize minute, yet significant, differences.

The model training approach in this experiment centered around the CLIP image encoder, enhanced with an additional fully connected layer. FACILE-FSP and FACILE-SupCon incorporate this setup, utilizing the CLIP-based feature encoder and focusing on finetuning the fully connected layer through the FACILE pretraining step. In contrast, the SimSiam approach leverages the CLIP image encoder as a backbone while finetuning the projector and predictor components. Similarly, the SimCLR method also uses the CLIP encoder as its foundation but focuses solely on finetuning the projector. These varied strategies reflect our efforts to optimize the feature encoder for accurately identifying anomalies and improving classification performance in related tasks. The training details can be found in §A.1.2.

Note that Table 5.2 clearly demonstrates that all models tested benefit from incorporating data from the target domain. Notably, both FACILE-SupCon and FACILE-FSP exhibit superior performance compared to other baseline models. This observation underscores the effectiveness of our models in leveraging coarse-grained labels to enhance their anomaly detection capabilities.

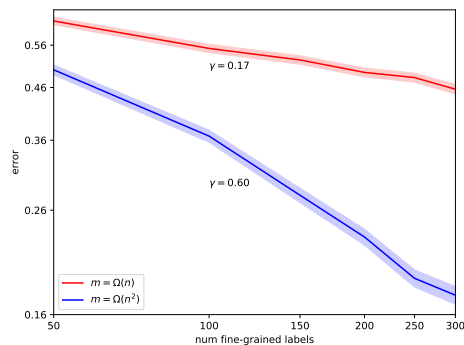


Figure 5.4: Generalization error (with two growth rates) of FACILE-FSP on CIFAR-100 test dataset as a function of the number of coarse-grained labels  $m$ .



pretraining method	NC	LR	RC
CLIP (ViT-B/16)	$83.84 \pm 1.10$	$81.01 \pm 1.23$	$82.75 \pm 1.17$
SimCLR	$84.03 \pm 1.08$	$83.49 \pm 1.14$	$86.30 \pm 1.03$
SimSiam	$84.02 \pm 1.10$	$83.90 \pm 1.13$	$85.68 \pm 1.07$
FACILE-SupCon	$87.49 \pm 0.99$	$86.57 \pm 1.07$	$88.01 \pm 0.99$
FACILE-FSP	<b><math>88.74 \pm 0.94</math></b>	<b><math>88.45 \pm 0.96</math></b>	<b><math>88.36 \pm 0.95</math></b>

Table 5.2: Pretraining on input sets from CUB200. Testing with 5-shot 20-way meta-test sets; average F1 and CI are reported.

### 5.3.5 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.,  $224 \times 224$  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  $224 \times 224$  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 §A.3.

We test 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 §A.3. Note that the TCGA and GTEx have meticulously categorized an extensive array of cancer types and organs, covering a diverse range of tissues as outlined in the LC, PAIP, and NCT. The strategic use of WSI-level labels is rooted in their potential

to enrich tissue-level classification. While these labels may appear broad, they encapsulate a wealth of underlying heterogeneity inherent to different cancer regions and tissue types.

**Pretrain ResNet18 on TCGA with patch size  $224 \times 224$**  We first train models on TCGA patches with size  $224 \times 224$  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 §A.1.3. 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 §A.5 for details of LA.

The test result is shown in Table 5.3. 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 5.3 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 down-

stream tasks. More experiment results about test ACC on LC, PAIP, and NCT datasets

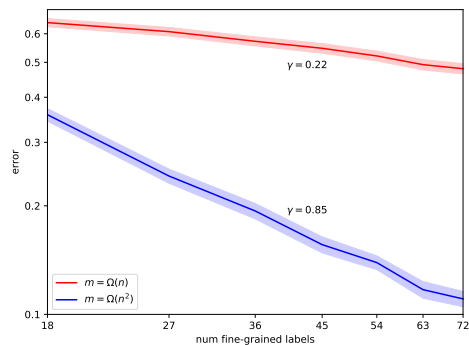


Figure 5.5: Generalization error on NCT dataset. The FACILE-FSP trains on TCGA dataset with  $m$  coarse-grained labels. We show the error curve with two growth rates of  $m$ .

can be found in §A.2.1. Test result with larger shot number is in §A.2.1. We further pre-train 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 §A.2.3 for experiment results on the PDAC dataset.

We show the generalization error of FACILE-FSP w.r.t. fine-grained label number in Figure. 5.5. The figure reveals a pronounced log-linear relationship. A larger growth rate of coarse-grained labels implies a faster rate of excess risk.

**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 §A.2.2. The test result on the LC dataset is shown in Table 5.4.

We can see from Table 5.4 the best model pretrained on NCT, i.e., FSP with strong augmentation, performs worse than our model FACILE-FSP in Table 5.3. Our method gets 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. A.3. Note that SimSiam model, trained with a batch size of 55, maintains competitive performance to MoCo v3 which needs a large batch size.

**More experiments and ablation study** We refer interested readers to §A.6 for ablation studies about set size, training procedures, and set-input models. These experiments extend our analysis to specialized tasks, showcasing the adaptability of FACILE to foundation models.

### 5.3.6 *Fine-tune ViT-B/14 of DINO V2 on TCGA Dataset*

Similar to §5.3.4, we fine-tune a fully connected layer that is appended after DINO V2 Oquab et al. [2023] ViT-B/14. This methodology is applied across various models to assess their performance on histopathology image datasets. By adopting the DINO V2 trained models, known for their robustness and effectiveness in visual representation learning, we aim to harness their potential for the specialized domain of histopathology. We refer interested readers to §A.1.4 for details of pretraining.

Notably, our methods, FACILE-SupCon and FACILE-FSP, demonstrated markedly superior results in comparison to other baseline models when applied to histopathology image datasets as shown in Table 5.5. This outcome highlights the effectiveness of these methods in leveraging coarse-grained labels specific to histopathology, thereby greatly enhancing the model performance of downstream tasks. Another critical insight emerged from our research: the current foundation model, DINO V2, exhibits limitations in its generalization performance on histopathology images. This suggests that while DINO V2 provides a strong starting point due to its robust visual representation capabilities, there is a clear need for further finetuning or prompt learning to optimize its performance for the unique challenges presented by histopathology datasets. This finding underscores the importance of specialized adaptation in the application of foundation models to specific domains like medical imaging.

## 5.4 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 and 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 Wu et al. [2018a], Phoo and 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 via a hierarchy. Also, 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.

## 5.5 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 labels to benefit 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. It is important to note that the separate utilization of loosely related coarse-grained labels and fine-grained labels can be expensive. Specifically, the pretraining of our proposed algorithm could be expensive given large amounts of coarse-grained data and the nature of the set-input data. For this reason, we are investigating methods of selecting a subset of the coarse-grained dataset to accelerate pretraining.

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	74.91 ± 1.25	<b>76.23 ± 1.16</b>	75.01 ± 1.19	75.60 ± 1.19	<b>75.64 ± 1.18</b>
FACILE-FSP	<b>77.39 ± 1.21</b>	76.14 ± 1.25	<b>75.18 ± 1.30</b>	<b>77.55 ± 1.17</b>	73.72 ± 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	91.09 ± 0.47	90.34 ± 0.48	90.25 ± 0.48	91.32 ± 0.47	<b>90.94 ± 0.50</b>
FACILE-FSP	<b>91.67 ± 0.45</b>	<b>90.64 ± 0.50</b>	<b>90.52 ± 0.52</b>	<b>92.07 ± 0.48</b>	89.81 ± 0.61
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	<b>48.85 ± 1.21</b>	<b>49.44 ± 1.26</b>	<b>50.27 ± 1.22</b>	<b>49.76 ± 1.20</b>	<b>48.44 ± 1.21</b>
FACILE-SupCon	46.60 ± 1.20	48.63 ± 1.22	48.46 ± 1.21	47.13 ± 1.20	47.87 ± 1.22
FACILE-FSP	45.40 ± 1.24	46.71 ± 1.20	46.60 ± 1.21	46.36 ± 1.22	45.49 ± 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	<b>64.74 ± 0.91</b>	<b>65.63 ± 0.97</b>	<b>65.93 ± 0.97</b>	66.68 ± 0.86	<b>66.48 ± 0.82</b>
FACILE-FSP	63.90 ± 0.94	64.59 ± 0.96	65.43 ± 0.96	<b>66.77 ± 0.86</b>	66.34 ± 0.85
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	<b>68.10 ± 1.29</b>	<b>69.63 ± 1.25</b>	<b>69.81 ± 1.24</b>	<b>69.54 ± 1.25</b>	<b>69.77 ± 1.22</b>
FACILE-FSP	66.38 ± 1.38	67.03 ± 1.34	67.56 ± 1.32	68.35 ± 1.33	<b>69.77 ± 1.30</b>
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	<b>86.79 ± 0.61</b>	<b>87.89 ± 0.58</b>	<b>89.10 ± 0.52</b>	<b>89.53 ± 0.52</b>	<b>88.58 ± 0.54</b>
FACILE-FSP	84.68 ± 0.74	85.47 ± 0.72	87.44 ± 0.64	88.00 ± 0.63	87.51 ± 0.66

Table 5.3: 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)	<b>83.53 ± 0.79</b>	<b>80.81 ± 1.01</b>	<b>80.27 ± 1.08</b>	<b>85.57 ± 0.77</b>	<b>84.06 ± 0.89</b>
SupCon	81.51 ± 0.85	78.77 ± 1.03	78.65 ± 1.08	83.51 ± 0.84	83.31 ± 0.91

Table 5.4: Pretraining on NCT and 5-shot 5-way testing on LC dataset; average F1 and CI are reported.



pretraining method	NC	LR	RC	LR+LA	RC+LA
1-shot 5-way test on LC dataset					
DINO V2 (ViT-B/14)	44.82 ± 1.41	47.51 ± 1.39	47.63 ± 1.38	47.36 ± 1.39	48.88 ± 1.44
SimSiam	48.79 ± 1.37	49.43 ± 1.35	48.43 ± 1.36	49.38 ± 1.34	49.50 ± 1.34
SimCLR	50.47 ± 1.31	50.52 ± 1.33	50.44 ± 1.32	51.66 ± 1.32	51.78 ± 1.38
FSP-Patch	49.73 ± 1.41	53.59 ± 1.38	53.07 ± 1.41	51.79 ± 1.40	51.27 ± 1.43
FACILE-SupCon	<b>56.24 ± 1.43</b>	<b>56.51 ± 1.41</b>	<b>55.95 ± 1.42</b>	<b>56.29 ± 1.43</b>	54.07 ± 1.44
FACILE-FSP	55.67 ± 1.40	56.26 ± 1.36	55.83 ± 1.35	56.01 ± 1.38	<b>55.35 ± 1.40</b>
5-shot 5-way test on LC dataset					
DINO V2 (ViT-B/14)	66.12 ± 0.98	64.71 ± 1.12	66.36 ± 1.10	72.95 ± 0.93	75.11 ± 0.91
SimSiam	67.51 ± 0.96	64.99 ± 1.05	65.39 ± 1.05	70.30 ± 0.93	71.19 ± 0.93
SimCLR	70.10 ± 0.92	69.28 ± 0.96	69.18 ± 0.97	72.99 ± 0.92	72.91 ± 0.94
FSP-Patch	71.97 ± 0.96	71.11 ± 1.04	71.19 ± 1.03	73.96 ± 0.94	73.20 ± 0.96
FACILE-SupCon	75.58 ± 0.88	74.26 ± 0.94	73.20 ± 0.95	75.81 ± 0.90	74.34 ± 0.96
FACILE-FSP	<b>75.86 ± 0.86</b>	<b>74.64 ± 0.89</b>	<b>74.12 ± 0.93</b>	<b>76.17 ± 0.88</b>	<b>75.08 ± 0.95</b>
1-shot 3-way test on PAIP dataset					
DINO V2 (ViT-B/14)	41.51 ± 1.27	44.37 ± 1.26	44.28 ± 1.25	42.43 ± 1.27	42.78 ± 1.27
SimSiam	49.42 ± 1.28	48.07 ± 1.35	48.44 ± 1.36	48.76 ± 1.33	46.48 ± 1.37
SimCLR	48.60 ± 1.19	48.76 ± 1.25	47.98 ± 1.26	48.94 ± 1.23	47.20 ± 1.26
FSP-Patch	46.09 ± 1.17	47.44 ± 1.18	48.09 ± 1.19	46.76 ± 1.18	43.68 ± 1.22
FACILE-SupCon	<b>51.97 ± 1.18</b>	<b>52.25 ± 1.22</b>	<b>51.80 ± 1.22</b>	51.36 ± 1.22	<b>50.24 ± 1.23</b>
FACILE-FSP	51.34 ± 1.16	51.18 ± 1.19	51.51 ± 1.19	<b>51.50 ± 1.16</b>	49.77 ± 1.22
5-shot 3-way test on PAIP dataset					
DINO V2 (ViT-B/14)	57.59 ± 1.07	58.19 ± 1.10	59.37 ± 1.07	61.84 ± 0.85	60.81 ± 0.86
SimSiam	61.56 ± 0.97	62.52 ± 1.01	62.81 ± 1.01	64.40 ± 0.86	62.44 ± 0.93
SimCLR	62.20 ± 0.93	61.78 ± 0.99	63.20 ± 0.97	63.38 ± 0.86	63.03 ± 0.88
FSP-Patch	63.77 ± 0.88	63.85 ± 0.94	63.85 ± 0.93	63.61 ± 0.85	60.91 ± 0.87
FACILE-SupCon	<b>67.16 ± 0.84</b>	67.29 ± 0.89	66.88 ± 0.90	<b>67.61 ± 0.85</b>	<b>66.34 ± 0.84</b>
FACILE-FSP	67.14 ± 0.85	<b>67.67 ± 0.84</b>	<b>67.54 ± 0.86</b>	67.12 ± 0.81	66.05 ± 0.83
1-shot 9-way test on NCT dataset					
DINO V2 (ViT-B/14)	56.03 ± 1.62	59.11 ± 1.57	60.13 ± 1.55	58.71 ± 1.57	59.06 ± 1.55
SimSiam	62.60 ± 1.45	61.89 ± 1.50	61.90 ± 1.51	62.27 ± 1.47	61.05 ± 1.44
SimCLR	65.43 ± 1.43	64.18 ± 1.44	64.15 ± 1.46	64.83 ± 1.43	62.69 ± 1.38
FSP-Patch	65.22 ± 1.49	65.93 ± 1.41	65.94 ± 1.40	65.26 ± 1.45	62.66 ± 1.46
FACILE-SupCon	71.55 ± 1.36	70.36 ± 1.37	70.52 ± 1.35	71.05 ± 1.35	<b>68.85 ± 1.40</b>
FACILE-FSP	<b>72.05 ± 1.34</b>	<b>70.70 ± 1.35</b>	<b>70.77 ± 1.34</b>	<b>71.14 ± 1.34</b>	68.03 ± 1.40
5-shot 9-way test on NCT dataset					
DINO V2 (ViT-B/14)	76.85 ± 0.98	76.51 ± 1.02	78.67 ± 0.94	82.20 ± 0.82	82.75 ± 0.83
SimSiam	80.81 ± 0.85	80.06 ± 0.87	81.55 ± 0.85	83.18 ± 0.80	82.39 ± 0.83
SimCLR	82.87 ± 0.80	81.91 ± 0.82	82.86 ± 0.80	83.92 ± 0.77	82.89 ± 0.79
FSP-Patch	83.63 ± 0.83	83.49 ± 0.80	84.34 ± 0.78	85.32 ± 0.75	83.03 ± 0.79
FACILE-SupCon	87.74 ± 0.64	87.00 ± 0.64	87.38 ± 0.62	87.82 ± 0.63	86.15 ± 0.69
FACILE-FSP	<b>87.93 ± 0.65</b>	<b>87.52 ± 0.65</b>	<b>87.72 ± 0.62</b>	<b>88.01 ± 0.64</b>	<b>86.46 ± 0.70</b>

Table 5.5: Test result on LC, PAIP, and NCT dataset with ViT-B/14 from DINO V2; average F1 and CI are reported.

# CHAPTER 6

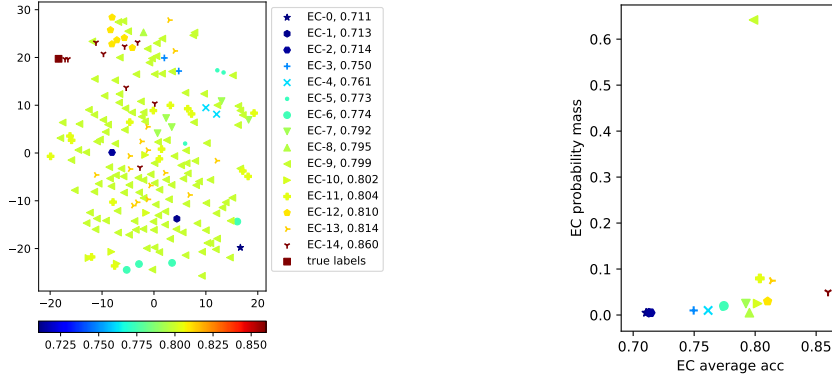
## DEEP BAYESIAN ACTIVE LEARNING

### 6.1 Motivation

Active learning (AL) [Settles, 2012] characterizes a collection of techniques that efficiently select data for training machine learning models. In the *pool-based* setting, an active learner selectively queries the labels of data points from a pool of unlabeled examples and incurs a certain cost for each label obtained. The goal is to minimize the total cost while achieving a target level of performance. A common practice for AL is to devise efficient surrogates, aka *acquisition functions*, to assess the effectiveness of unlabeled data points in the pool.

There has been a vast body of literature and empirical studies [Huang et al., 2010, Houlsby et al., 2011, Wang and Ye, 2015, Hsu and Lin, 2015, Huang et al., 2016, Sener and Savarese, 2017, Ducoffe and Precioso, 2018, Ash et al., 2019, Liu et al., 2020, Yan et al., 2020] suggesting a variety of heuristics as potential acquisition functions for AL. Among these methods, *Bayesian Active Learning by Disagreement* (BALD) [Houlsby et al., 2011] has attained notable success in the context of deep Bayesian AL, while maintaining the expressiveness of Bayesian models [Gal et al., 2017, Janz et al., 2017, Shen et al., 2017]. Concretely, BALD relies on a *most informative selection* (MIS) strategy—a classical heuristic that dates back to Lindley [1956]—which greedily queries the data point exhibiting the maximal *mutual information* with the model parameters at each iteration. Despite the overwhelming popularity of such heuristics due to the algorithmic simplicity [MacKay, 1992, Chen et al., 2015b, Gal and Ghahramani, 2016], the performance of these AL algorithms, unfortunately, is *sensitive* to the quality of uncertainty estimations of the underlying model, and it remains an open problem in deep AL to accurately quantify the model uncertainty, due to limited access to training data and the challenge of posterior estimation.

In Figure. 6.1, we demonstrate the potential issues of MIS-based strategies introduced



(a) Samples from posterior BNN via MC (b) Probability mass (y-axis) of equivalence dropout. classes.

Figure 6.1: (a) The embeddings are generated by applying t-SNE on the hypotheses’ predictions on a random hold-out dataset. The colorbar indicates the (approximate) test accuracy of the sampled neural networks on the MNIST dataset. See §B.1 for details of the experimental setup. (b) Probability mass (y-axis) of equivalence classes (sorted by the average accuracy of the enclosed hypotheses as the x-axis).

by inaccurate posterior samples from a Bayesian Neural Network (BNN) on a multi-class classification dataset. Here, the samples (i.e. hypotheses) from the model posterior are grouped into *equivalence classes* (ECs) [Golovin et al., 2010] according to the Hamming distance between their predictions as shown in Figure. 6.1a. Informally, an equivalence class contains hypotheses that are close in their predictions for a randomly selected set of examples. We note from Figure. 6.1a that the probability mass of the models sampled from the BNN is centered around the mode of the approximate posterior distribution, while little coverage is seen on models of higher accuracy. Consequently, MIS tends to select data points that reveal the maximal information w.r.t. the *sampled distribution*, rather than guiding the active learner towards learning high accuracy models.

In addition to the *robustness* concern, another challenge for deep AL is the *scalability* to large batches of queries. In many real-world applications, fully sequential data acquisition algorithms are often undesirable especially for large models, as model retraining becomes the bottleneck of the learning system [Mittal et al., 2019, Ostapuk et al., 2019]. Due to

such concerns, batch-mode algorithms are designed to reduce the computational time spent on model retraining and increase labeling efficiency. Unfortunately, for most acquisition functions, computing the optimal batch of queries function is NP-hard [Chen and Krause, 2013a]; when the evaluation of the acquisition function is expensive or the pool of candidate queries is large, it is even computationally challenging to construct a batch greedily [Gal et al., 2017, Kirsch et al., 2019, Ash et al., 2019]. Recently, efforts in scaling up batch-mode AL algorithms often involve diversity sampling strategies [Sener and Savarese, 2017, Ash et al., 2019, Citovsky et al., 2021, Kirsch et al., 2021a]. Unfortunately, these diversity selection strategies either ignore the downstream learning objective (e.g., using clustering as by [Citovsky et al., 2021]) or inherit the limitations of the sequential acquisition functions (e.g., sensitivity to uncertainty estimate as elaborated in Figure. 6.1 [Kirsch et al., 2021a]).

Motivated by these two challenges, this chapter aims to simultaneously (1) mitigate the limitations of uncertainty-based deep AL heuristics due to inaccurate uncertainty estimation, and (2) enable efficient computation of batches of queries at scale. Parts of this chapter are replicated from Zhang et al. [2023b] with some modifications.

We propose Batch-BALANCE—an efficient batch-mode deep Bayesian AL framework—which employs a decision-theoretic acquisition function inspired by Golovin et al. [2010], Chen et al. [2016]. Concretely, Batch-BALANCE utilizes BNNs as the underlying hypotheses and uses Monte Carlo (MC) dropout [Gal and Ghahramani, 2016, Kingma et al., 2015] or Stochastic gradient Markov Chain Monte Carlo (SG-MCMC) [Welling and Teh, 2011, Chen et al., 2014, Ding et al., 2014, Li et al., 2016a] to estimate the model posterior. It then selects points that can most effectively tell apart hypotheses from different equivalence classes (as illustrated in Figure. 6.1). Intuitively, such disagreement structure is induced by the pool of unlabeled data points; therefore our selection criterion takes into account the informativeness of a query with respect to the target models (as done in BALD) while putting less focus on differentiating models with little disagreement on target data distribution. As learning

progresses, Batch-BALANCE adaptively anneals the radii of the equivalence classes, resulting in selecting more “difficult examples” that distinguish more similar hypotheses as the model accuracy improves.

When computing queries in small batches, Batch-BALANCE employs an importance sampling strategy to efficiently compute the expected gain in differentiating equivalence classes for a batch of examples and chooses samples within a batch in a greedy manner. To scale up the computation of queries to large batches, we further propose an efficient batch-mode acquisition procedure, which aims to maximize a novel *combinatorial information measure* [Kothawade et al., 2021] defined through our novel acquisition function. The resulting algorithm can efficiently scale to realistic batched learning tasks with reasonably large batch sizes.

Finally, we demonstrate the effectiveness of variants of Batch-BALANCE via an extensive empirical study, and show that they achieve compelling performance—sometimes by a large margin—on several benchmark datasets.

## 6.2 Problem Setup

### 6.2.1 Problem Statement

**Notations** We consider pool-based Bayesian AL, where we are given an unlabelled dataset  $\mathcal{D}_{\text{pool}}$  drawn *i.i.d.* from some underlying data distribution. Further, assume a labeled dataset  $\mathcal{D}_{\text{train}}$  and a set of hypotheses  $\mathcal{H} = \{h_1, \dots, h_n\}$ . We would like to distinguish a set of (unknown) target hypotheses among the ground set of hypotheses  $\mathcal{H}$ . Let  $H$  denote the random variable that represents the target hypotheses. Let  $p(H)$  be a prior distribution over the hypotheses. In this paper, we resort to BNN with parameters  $\omega \sim p(\omega \mid \mathcal{D}_{\text{train}})$ <sup>1</sup>.

---

1. We use the conventional notation  $\omega$  to represent the parameters of a BNN and use  $\omega$  and  $h$  interchangeably to denote a hypothesis.

**Problem Statement** An AL algorithm will select samples from  $\mathcal{D}_{\text{pool}}$  and query labels from experts. The experts will provide label  $y$  for given query  $x \in \mathcal{D}_{\text{pool}}$ . We assume labeling each query  $x$  incurs a unit cost.

Our goal is to find an adaptive policy for selecting samples that allows us to find a hypothesis with a target error rate  $\sigma \in [0, 1]$  while minimizing the total cost of the queries. Formally, a *policy*  $\pi$  is a mapping  $\pi$  from the labeled dataset  $\mathcal{D}_{\text{train}}$  to samples in  $\mathcal{D}_{\text{pool}}$ .

We use  $\mathcal{D}_{\text{train}}^\pi$  to denote the set of examples chosen by  $\pi$ . Given the labeled dataset  $\mathcal{D}_{\text{train}}^\pi$ , we define  $p_{\text{ERR}}(\pi)$  as the expected error probability w.r.t. the posterior  $p(\omega \mid \mathcal{D}_{\text{train}}^\pi)$ . Let the cost of a policy  $\pi$  be  $\text{cost}(\pi) \triangleq \max | \mathcal{D}_{\text{train}}^\pi |$ , i.e., the maximum number of queries made by policy  $\pi$  over all possible realizations of the target hypothesis  $H \in \mathcal{H}$ . Given a tolerance parameter  $\sigma \in [0, 1]$ , we seek a policy with minimal cost, such that upon termination, it will get an expected error probability less than  $\sigma$ . Formally, we seek  $\arg \min_{\pi} \text{cost}(\pi)$ , s.t.  $p_{\text{ERR}}(\pi) \leq \sigma$ .

### 6.2.2 Most Informative Selection Criterion

BALD uses mutual information between the model prediction for each sample and the parameters of the model as the acquisition function. It captures the reduction of model uncertainty by receiving a label  $y$  of a data point  $x$ :

$\mathbb{I}(y; \omega \mid x, \mathcal{D}_{\text{train}}) = \mathbb{H}(y \mid x, \mathcal{D}_{\text{train}}) - \mathbb{E}_{p(\omega \mid \mathcal{D}_{\text{train}})} [\mathbb{H}(y \mid x, \omega, \mathcal{D}_{\text{train}})]$  where  $\mathbb{H}$  denotes the Shannon entropy [Shannon, 1948]. Kirsch et al. [2019] further proposed BatchBALD as an extension of BALD whereby the mutual information between a joint of multiple data points and the model parameters is estimated as

$$\Delta_{\text{BatchBALD}}(x_{1:b} \mid \mathcal{D}_{\text{train}}) \triangleq \mathbb{I}(y_{1:b}; \omega \mid x_{1:b}, \mathcal{D}_{\text{train}}).$$

**Limitation of the BALD algorithm** BALD can be ineffective when the hypothesis samples are heavily biased and cluttered towards sub-optimal hypotheses. Below, we provide a concrete example where such selection criteria may be undesirable.

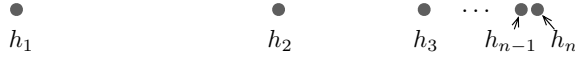


Figure 6.2: A stylized example where the most informative selection criterion underperforms the equivalence-class-based criterion.

Consider the problem shown in Figure. 6.2. The hypothesis class  $\mathcal{H} = \{h_1, \dots, h_n\}$  is structured such that

$$d_{\mathbb{H}}(h_i, h_j) = \begin{cases} 2^{1-i} - 2^{1-j} & \text{if } i < j, \\ 2^{1-j} - 2^{1-i} & \text{o.w.} \end{cases}$$

where  $d_{\mathbb{H}}(h_i, h_j)$  denotes the fraction of labels  $h_i$  and  $h_j$  disagree upon when making predictions on *i.i.d.* samples of data points. We further assume that for any subset of hypotheses  $S \subseteq \mathcal{H}$ , there exists a data point whose label they agree upon.

Assume each hypothesis  $h_i$  has an equal probability and the target error rate is  $\sigma$ . On the one hand, note that BALD does not consider  $d_{\mathbb{H}}(h_i, h_j)$ , and therefore on average it requires  $\log n$  examples to identify any target hypothesis. On the other hand, to achieve a target error rate of  $\sigma$ , one only needs to differentiate all pairs of hypotheses  $h_i, h_j$  of distance  $d_{\mathbb{H}}(h_i, h_j) > \sigma$  (i.e., by selecting training examples to rule out at least one of  $h_i, h_j$ ). Therefore, a “smarter” AL policy could query examples to sequentially check the consistency of  $h_1, h_2, \dots, h_n$  until all remaining hypotheses are within distance  $\sigma$ . It is easy to check that this requires  $\log(1/\sigma)$  examples before reaching the error rate  $\sigma$ . The gap between BALD and the above policy  $\frac{\log n}{\log(1/\sigma)}$  could be large as  $n$  increases.

### 6.2.3 Equivalence-class-based Selection Criterion

As alluded in §6.1 and Figure. 6.1, the MIS strategy can be ineffective when the samples from the model posterior are heavily biased and cluttered toward sub-optimal hypotheses.

A “smarter” strategy would instead leverage the structure of the hypothesis space induced by the underlying (unlabeled) pool of data points. In fact, this idea connects to an important problem for approximate AL, which is often cast as learning *equivalence classes* [Golovin et al., 2010]:

**Definition 1** (Equivalence Class). *Let  $(\mathcal{H}, d)$  be a metric space where  $\mathcal{H}$  is a hypothesis class and  $d$  is a metric. For a given set  $\mathcal{V} \subseteq \mathcal{H}$  and centers  $\mathcal{S} = \{s_1, \dots, s_k\} \subseteq \mathcal{V}$  of size  $k$ , let  $r^{\mathcal{S}} : \mathcal{V} \rightarrow [k]$  be a partition function over  $\mathcal{V}$  and  $\mathcal{D}_i := \{h \in \mathcal{V} \mid r^{\mathcal{S}}(h) = i\}$ , such that  $\forall i, j \in [k], r^{\mathcal{S}}(s_i) = i$  and  $\forall h \in \mathcal{D}_i, d(h, s_i) \leq d(h, s_j)$ . Each  $\mathcal{D}_i \subseteq \mathcal{V}$  is called an equivalence class induced by  $s_i \in \mathcal{S}$ .*

Consider a pool-based AL problem with hypothesis space  $\mathcal{H}$ , a sampled set  $\mathcal{V} \subseteq \mathcal{H}$ , and an unlabeled dataset  $\bar{\mathcal{D}}_{\text{pool}}$  which is drawn i.i.d. from the underlying data distribution. Each hypothesis  $h \in \mathcal{H}$  can be represented by a vector  $v_h$  indicating the predictions of all samples in  $\bar{\mathcal{D}}_{\text{pool}}$ . We can construct equivalence classes with the Hamming distance, which is denoted as  $d_{\mathbb{H}}(h, h')$ , and equivalence class number  $k$  on sampled hypotheses  $\mathcal{V}$ . Let  $d_{\mathbb{H}}^{\mathcal{S}}(\mathcal{V}) := \max_{h, h' \in \mathcal{V}: r^{\mathcal{S}}(h) = r^{\mathcal{S}}(h')} d_{\mathbb{H}}(h, h')$  be the maximal diameter of equivalence classes induced by  $\mathcal{S}$ .

Therefore, the error rates of any unordered pair of hypotheses  $\{h, h'\}$  that lie in the same equivalence class are at most  $d_{\mathbb{H}}^{\mathcal{S}}(\mathcal{V})$  away from each other. If we construct the  $k$  equivalence-class-inducing centers (as in Definition 1) as the solution of the max-diameter clustering problem:  $\mathcal{C} = \arg \min_{|\mathcal{S}|=k} d_{\mathbb{H}}^{\mathcal{S}}(\mathcal{V})$ , we can obtain the minimal worst-case relative error (i.e. difference in error rate) between hypotheses pair  $\{h, h'\}$  that lie in the same equivalence class.



We denote  $\mathcal{E} = \{\{h, h'\} : r^{\mathcal{C}}(h) \neq r^{\mathcal{C}}(h')\}$  as the set of all (unordered) pairs of hypotheses (i.e. undirected edges) corresponding to different equivalence classes with centers in  $\mathcal{C}$ .

**Equivalence class edge cutting** Consider the problem statement in §6.2.1. If  $\sigma = 0$  and tests are noise-free, this problem can be solved near-optimally by the *equivalence class edge cutting* (EC<sup>2</sup>) algorithm [Golovin et al., 2010]. EC<sup>2</sup> employs an edge-cutting strategy based on a weighted graph  $G = (\mathcal{H}, \mathcal{E})$ , where vertices represent hypotheses and edges link hypotheses that we want to distinguish between. Here  $\mathcal{E} \triangleq \{\{h, h'\} : r(h) \neq r(h')\}$  contains all pairs of hypotheses that have different equivalence classes. We define a weight function  $W : \mathcal{E} \rightarrow \mathbb{R}_{\geq 0}$  by  $W(\{h, h'\}) \triangleq p(h) \cdot p(h')$ . A sample  $x$  with label  $y$  is said to "cut" an edge if at least one hypothesis is inconsistent with  $y$ . Denote  $\mathcal{E}(x, y) \triangleq \{\{h, h'\} \in \mathcal{E} : p(y | x, h) = 0 \vee p(y | x, h') = 0\}$  as the set of edges cut by labeling  $x$  as  $y$ . The EC<sup>2</sup> objective is then defined as the total weight of edges cut by the current  $\mathcal{D}_{\text{train}}$ :  $f_{\text{EC}^2}(\mathcal{D}_{\text{train}}) \triangleq W\left(\bigcup_{(x,y) \in \mathcal{D}_{\text{train}}} \mathcal{E}(x, y)\right)$ . EC<sup>2</sup> algorithm greedily maximizes this objective per iteration. The acquisition function for EC<sup>2</sup> is

$$\Delta_{\text{EC}^2}(x | \mathcal{D}_{\text{train}}) \triangleq \mathbb{E}_y [f(\mathcal{D}_{\text{train}} \cup \{(x, y)\}) - f(\mathcal{D}_{\text{train}}) | \mathcal{D}_{\text{train}}]. \quad (6.1)$$

**The equivalence class edge discounting algorithm** In the noisy setting, the acquisition function of *Equivalence Class Edge Discounting* algorithm (ECED) [Chen et al., 2016] takes undesired contribution by noise into account. Given a data point and its label  $(x, y)$ ,

ECED discounts all model parameters by their likelihood ratio:  $\lambda_{h,y} \triangleq \frac{p(y|h,x)}{\max_{y'} p(y'|h,x)}$ . After we get  $\mathcal{D}_{\text{train}}$ , the value of assigning label  $y$  to a data point  $x$  is defined as the total amount of edge weight discounted:  $\delta(x, y | \mathcal{D}_{\text{train}}) \triangleq \sum_{\{h,h'\} \in \mathcal{E}} p(h, \mathcal{D}_{\text{train}}) p(h', \mathcal{D}_{\text{train}}) \cdot (1 - \lambda_{h,y} \lambda_{h',y})$ , where  $\mathcal{E} = \{\{h, h'\} : r(h) \neq r(h')\}$  consists of all unordered pairs of hypothesis corresponding to different equivalence classes. Further, ECED augments the above value function  $\delta$  with an offset value such that the value of a non-informative test is 0. The offset

value of labeling  $x$  as label  $y$  is defined as:  $\nu(x, y | \mathcal{D}_{\text{train}}) \triangleq \sum_{\{h, h'\} \in \mathcal{E}} p(h, \mathcal{D}_{\text{train}}) p(h', \mathcal{D}_{\text{train}}) \cdot (1 - \max_h \lambda_{h,y}^2)$ . The overall acquisition function of ECED is:

$$\Delta_{\text{ECED}}(x | \mathcal{D}_{\text{train}}) \triangleq \mathbb{E}_y [\delta(x, y | \mathcal{D}_{\text{train}}) - \nu(x, y | \mathcal{D}_{\text{train}})]. \quad (6.2)$$

**Limitation of existing EC-based algorithms** Existing EC-based AL algorithms (e.g., EC<sup>2</sup> [Golovin et al., 2010] and ECED [Chen et al., 2016]) are not directly applicable to deep Bayesian AL tasks. This is because computing the acquisition function (i.e., Eq. (6.1) and Eq. (6.2)) needs to integrate over the hypotheses space, which is intractable for large models (such as deep BNN). Moreover, it is nontrivial to extend to batch-mode setting since the number of possible candidate batches and the number of label configurations for the candidate batch grows exponentially with the batch size. Therefore, we need efficient approaches to approximate the ECED acquisition function when dealing with BNNs in both fully sequential setting and batch-mode setting.

## 6.3 Our Approach

We first introduce our acquisition function for the sequential setting, namely BALANCE (as in Bayesian Active Learning via Equivalence Class Annealing), and then present the batch-mode extension under both small and large batch-mode AL settings.

### 6.3.1 The BALANCE Acquisition Function

We resort to the Monte Carlo method to estimate the acquisition function. Given all available labeled samples  $\mathcal{D}_{\text{train}}$  at each iteration, hypotheses  $\omega$  are sampled from the BNN posterior. We instantiate our methods with two different BNN posterior sampling approaches: MC dropout [Gal and Ghahramani, 2016] and cSG-MCMC [Zhang et al., 2019]. MC dropout is easy to implement and scales well to large models and datasets very efficiently [Kirsch et al.,

2019, Gal and Ghahramani, 2016, Gal et al., 2017]. However, it is often poorly calibrated [Foong et al., 2020, Fortuin et al., 2021]. cSG-MCMC is more practical and indeed has high-fidelity to the true posterior [Zhang et al., 2019, Fortuin et al., 2021, Wenzel et al., 2020].

In order to determine if there is an edge  $\{\hat{\omega}, \hat{\omega}'\}$  that connects a pair of sampled hypotheses  $\hat{\omega}, \hat{\omega}'$  (i.e., if they are in different equivalence classes), we calculate the Hamming distance  $d_{\text{H}}(\hat{\omega}, \hat{\omega}')$  between the predictions of  $\hat{\omega}, \hat{\omega}'$  on the unlabeled dataset  $\bar{\mathcal{D}}_{\text{pool}}$ . If the distance is greater than some threshold  $\tau$ , we consider the edge  $\{\hat{\omega}, \hat{\omega}'\} \in \hat{\mathcal{E}}$ ; otherwise not. We define the acquisition function of BALANCE for a set  $x_{1:b} \triangleq \{x_1, \dots, x_b\}$  as:

$$\Delta_{\text{BALANCE}}(x_{1:b} \mid \mathcal{D}_{\text{train}}) \triangleq \mathbb{E}_{y_{1:b}} \mathbb{E}_{\omega, \omega' \sim p(\omega \mid \mathcal{D}_{\text{train}})} \mathbb{1}_{d_{\text{H}}(\omega, \omega') > \tau} \cdot \left(1 - \lambda_{\omega, y_{1:b}} \lambda_{\omega', y_{1:b}}\right) \quad (6.3)$$

where  $\lambda_{\omega, y_{1:b}} \triangleq \frac{p(y_{1:b} \mid \omega, x_{1:b})}{\max_{y'_{1:b}} p(y'_{1:b} \mid \omega, x_{1:b})}$  is the likelihood ratio<sup>2</sup> [Chen et al., 2016], and  $\mathbb{1}_{d_{\text{H}}(\hat{\omega}_k, \hat{\omega}'_k) > \tau}$  is the indicator function. We can adaptively anneal  $\tau$  by setting  $\tau$  proportional to BNN’s validation error rate  $\varepsilon$  in each AL iteration.

In practice, we cannot directly compute Eq. (6.3); instead we estimate it with sampled BNN posteriors: We first acquire  $K$  pairs of BNN posterior samples  $\{\hat{\omega}, \hat{\omega}'\}$ . The Hamming distances  $d_{\text{H}}(\hat{\omega}, \hat{\omega}')$  between these pairs of BNN posterior samples are computed. Next, we calculate the weight discount factor  $1 - \lambda_{\hat{\omega}_k, y_{1:b}} \lambda_{\hat{\omega}'_k, y_{1:b}}$  for each possible label  $y$  and each pair  $\{\hat{\omega}, \hat{\omega}'\}$  where  $d_{\text{H}}(\hat{\omega}, \hat{\omega}') > \tau$ . At last, we take the expectation of the discounted weight over all  $y_{1:b}$  configurations. In summary,  $\Delta_{\text{BALANCE}}(x_{1:b})$  is approximated as

$$\frac{1}{2K^2} \sum_{y_{1:b}} \sum_{k=1}^K (p(y_{1:b} \mid \hat{\omega}_k) + p(y_{1:b} \mid \hat{\omega}'_k)) \sum_{k=1}^K \mathbb{1}_{d_{\text{H}}(\hat{\omega}_k, \hat{\omega}'_k) > \tau} \left(1 - \lambda_{\hat{\omega}_k, y_{1:b}} \lambda_{\hat{\omega}'_k, y_{1:b}}\right). \quad (6.4)$$

$\mathcal{D}_{\text{train}}$  is omitted for simplicity of notations. Note that in our algorithms we never

---

2. The likelihood ratio is used here (instead of the likelihood) so that the contribution of “non-informative examples” (e.g.,  $p(y'_{1:b} \mid \omega, x_{1:b}) = \text{const} \forall y'_{1:b}, \omega$ ) is zeroed out.

explicitly construct equivalence classes on BNN posterior samples, due to the fact that (1) it is intractable to find the exact solution for the max-diameter clustering problem and (2) an explicit partitioning of the hypotheses samples tends to introduce “unnecessary” edges where the incident hypotheses are closeby (e.g., if a pair of hypotheses lie on the adjacent edge between two hypothesis partitions), and therefore may overly estimate the utility of a query. Nevertheless, we conducted an empirical study of a variant of BALANCE with explicit partitioning (which underperforms BALANCE). We defer detailed discussion on this approach, as well as empirical study, to the

---

**Algorithm 2** Active selection w/ Batch-BALANCE

---

- 1: **input:**  $\mathcal{D}_{\text{pool}}, \bar{\mathcal{D}}_{\text{pool}}$ , acquisition batch size  $B$ , coldness parameter  $\beta$ , threshold  $\tau$ , and downsampling subset size  $|\mathcal{C}|$ .
  - 2: draw  $K$  random pairs of BNN posterior samples  $\{\hat{\omega}_k, \hat{\omega}'_k\}_{k=1}^K$
  - 3: **if**  $B$  is sufficiently small (see §6.4.2) **then**
  - 4:    $\mathcal{A}_B \leftarrow \text{GreedySelection}(\mathcal{D}_{\text{pool}}, \bar{\mathcal{D}}_{\text{pool}}, \{\hat{\omega}_k, \hat{\omega}'_k\}_{k=1}^K, \tau, B)$  {see §6.3.2}
  - 5: **else**
  - 6:   downsample subset  $\mathcal{C} \subset \mathcal{D}_{\text{pool}}$  with  $p(x) \sim \Delta_{\text{BALANCE}}(x)^\beta$
  - 7:    $\mathcal{S}_{1:B}, \mu_{1:B} \leftarrow \text{BALANCE-Clustering}(\mathcal{C}, \bar{\mathcal{D}}_{\text{pool}}, \{\hat{\omega}_k, \hat{\omega}'_k\}_{k=1}^K, \tau, \beta, B)$  {see §6.3.3}
  - 8:    $\mathcal{A}_B \leftarrow \mu_{1:B}$
  - 9: **output:**  $\mathcal{A}_B$
- 

In the fully sequential setting, we choose one sample  $x$  with top  $\Delta_{\text{BALANCE}}(x)$  in each AL iteration. In the batch-mode setting, we consider two strategies for selecting samples within a batch: greedy selection strategy for small batches and acquisition-function-driven clustering strategy for large batches. We refer to our full algorithm as Batch-BALANCE (algorithm 2) and expand on the batch-mode extensions in the following two subsections.

### 6.3.2 Greedy Selection Strategy

To avoid the combinatorial explosion of possible batch number, the greedy selection strategy selects sample  $x$  with maximum  $\Delta_{\text{BALANCE}}(x_{1:b-1} \cup \{x\})$  in the  $b$ -th step of a batch. However, the configuration  $y_{1:b}$  of a subset  $x_{1:b}$  expands exponentially with subset size  $b$ . In

order to efficiently estimate  $\Delta_{\text{BALANCE}}(x_{1:b})$ , we employ an importance sampling method. The current  $M$  configuration samples of  $y_{1:b}$  are drawn by concatenating previous drawn  $M$  samples of  $y_{1:b-1}$  and  $M$  samples of  $y_b$  (samples drawn from proposal distribution). The pseudocode for the greedy selection strategy is provided in algorithm 3.

---

**Algorithm 3** Greedy selection

---

- 1: **input:** a set of samples  $\mathcal{D}$ ,  $\bar{\mathcal{D}}_{\text{pool}}$ ,  $\{\hat{\omega}_k, \hat{\omega}'_k\}_{k=1}^K$ , threshold  $\tau$ , and  $B$
  - 2:  $\mathcal{A}_0 = \emptyset$
  - 3: **for**  $b \in [B]$  **do**
  - 4:   **for all**  $x \in \mathcal{D} \setminus \mathcal{A}_{b-1}$  **do**
  - 5:      $s_x \leftarrow \Delta_{\text{BALANCE}}(\mathcal{A}_{b-1} \cup \{x\})$
  - 6:    $x_b \leftarrow \arg \max_{x \in \mathcal{D} \setminus \mathcal{A}_{b-1}} s_x$
  - 7:    $\mathcal{A}_b \leftarrow \mathcal{A}_{b-1} \cup \{x_b\}$
  - 8: **output:** batch  $\mathcal{A}_B = \{x_1, \dots, x_B\}$
- 

**Importance sampling of configurations** When  $b$  becomes large, it is infeasible to enumerate all label configurations  $y_{1:b}$ . We use  $M$  MC samples of  $y_{1:b}$  to estimate the acquisition function and importance sampling to further reduce the computational time<sup>3</sup>. Given that  $p(y_{1:b} \mid \omega)$  can be factorized as  $p(y_{1:b-1} \mid \omega) \cdot p(y_b \mid \omega)$ , the acquisition function can be written as:

$$\begin{aligned}
& \Delta_{\text{Batch-BALANCE}}(x_{1:b} \mid \mathcal{D}_{\text{train}}) \\
& \triangleq \mathbb{E}_{y_{1:b}} \left[ \mathbb{E}_{p(\omega \mid \mathcal{D}_{\text{train}})} \mathbb{1}_{d_{\text{H}}(\omega_k, \omega'_k) > \tau} \left( 1 - \lambda_{\omega, y_{1:b}} \lambda_{\omega', y_{1:b}} \right) \right] \\
& = \mathbb{E}_{p(\omega \mid \mathcal{D}_{\text{train}})} \mathbb{E}_{p(y_{1:b} \mid \omega)} \left[ \mathbb{E}_{\omega, \omega' \sim p(\omega \mid \mathcal{D}_{\text{train}})} \mathbb{1}_{d_{\text{H}}(\omega_k, \omega'_k) > \tau} \left( 1 - \lambda_{\omega, y_{1:b}} \lambda_{\omega', y_{1:b}} \right) \right] \\
& = \mathbb{E}_{p(\omega \mid \mathcal{D}_{\text{train}})} \mathbb{E}_{p(y_{1:b-1} \mid \omega)} \mathbb{E}_{p(y_b \mid \omega)} \left[ \mathbb{E}_{\omega, \omega' \sim p(\omega \mid \mathcal{D}_{\text{train}})} \mathbb{1}_{d_{\text{H}}(\omega_k, \omega'_k) > \tau} \left( 1 - \lambda_{\omega, y_{1:b}} \lambda_{\omega', y_{1:b}} \right) \right]
\end{aligned}$$

---

3. A similar importance sampling procedure was proposed in Kirsch et al. [2019] to estimate the mutual information. Here, we show how one can adapt the strategy to enable efficient estimation of  $\Delta_{\text{Batch-BALANCE}}$ .

Suppose we have  $M$  samples of  $y_{1:b-1}$  from  $p(y_{1:b-1})$ , we perform importance sampling using  $p(y_{1:b-1})$  to estimate the acquisition function:

$$\begin{aligned}
& \Delta_{\text{Batch-BALANCE}}(x_{1:b} \mid \mathcal{D}_{\text{train}}) \\
&= \mathbb{E}_{p(\omega \mid \mathcal{D}_{\text{train}})} \mathbb{E}_{p(y_{1:b-1})} \frac{p(y_{1:b-1} \mid \omega)}{p(y_{1:b-1})} \mathbb{E}_{p(y_b \mid \omega)} \left[ \mathbb{E}_{\omega, \omega' \sim p(\omega \mid \mathcal{D}_{\text{train}})} \mathbb{1}_{d_{\text{H}}(\omega, \omega') > \tau} \left( 1 - \lambda_{\omega, y_{1:b}} \lambda_{\omega', y_{1:b}} \right) \right] \\
&= \mathbb{E}_{p(y_{1:b-1})} \mathbb{E}_{p(\omega \mid \mathcal{D}_{\text{train}})} \mathbb{E}_{p(y_b \mid \omega)} \frac{p(y_{1:b-1} \mid \omega)}{p(y_{1:b-1})} \left[ \mathbb{E}_{\omega, \omega' \sim p(\omega \mid \mathcal{D}_{\text{train}})} \mathbb{1}_{d_{\text{H}}(\omega, \omega') > \tau} \left( 1 - \lambda_{\omega, y_{1:b}} \lambda_{\omega', y_{1:b}} \right) \right] \\
&\approx \frac{1}{M} \sum_{\hat{y}_{1:b-1}} \sum_{\hat{y}_b} \frac{\frac{1}{K} \sum_{k=1}^K p(\hat{y}_{1:b-1} \mid \hat{\omega}_k) p(\hat{y}_b \mid \hat{\omega}_k) + p(\hat{y}_{1:b-1} \mid \hat{\omega}'_k) p(\hat{y}_b \mid \hat{\omega}'_k)}{p(\hat{y}_{1:b-1})} \\
&\quad \left[ \frac{1}{K} \sum_{k=1}^K \mathbb{1}_{d_{\text{H}}(\hat{\omega}_k, \hat{\omega}'_k) > \tau} \left( 1 - \lambda_{\hat{\omega}_k, \hat{y}_{1:b}} \lambda_{\hat{\omega}'_k, \hat{y}_{1:b}} \right) \right] \\
&= \left( \frac{1}{K} \mathbb{1}_{d_{\text{H}}(\hat{\omega}_k, \hat{\omega}'_k) > \tau} \right)^\top \left( 1 - \frac{\hat{P}_{1:b-1} \otimes \hat{P}_b}{\hat{A}_{1:b}} \odot \frac{\hat{P}'_{1:b-1} \otimes \hat{P}'_b}{\hat{A}'_{1:b}} \right) \\
&\quad \left( \frac{1}{M} \frac{\hat{P}_{1:b-1}^\top \hat{P}_b + \hat{P}'_{1:b-1}^\top \hat{P}'_b}{\mathbb{1}^\top (\hat{P}_{1:b-1} + \hat{P}'_{1:b-1})} \right)^\top.
\end{aligned} \tag{6.5}$$

Here we save  $p(\hat{y}_{1:b-1} \mid \hat{\omega}_k)$  and  $p(\hat{y}_{1:b-1} \mid \hat{\omega}'_k)$  for  $M$  samples in  $\hat{P}_{1:b-1}$  and  $\hat{P}'_{1:b-1}$ . The shape of  $\hat{P}_{1:b-1}$  and  $\hat{P}'_{1:b-1}$  is  $K \times M$ .  $\odot$  is element-wise matrix multiplication and  $\otimes$  is the outer-product operator along first dimension. After the outer product operation, we can reshape the matrix by flattening all the dimensions after the 1st dimension.  $\mathbb{1}$  is a matrix of 1s with shape  $K \times 1$ .  $\hat{P}_{1:b-1}^\top \hat{P}_b$  and  $\hat{P}'_{1:b-1}^\top \hat{P}'_b$  are of shape  $M \times C$  and their sum is reshape to  $1 \times MC$  after divided by  $\mathbb{1}^\top (\hat{P}_{1:b-1} + \hat{P}'_{1:b-1})$ .

**Efficient implementation for greedy selection** In algorithm 3, we can store  $p(\hat{y}_{1:b-1} \mid \hat{\omega}_k)$  in a matrix  $\hat{P}_{1:b-1}$  and  $p(\hat{y}_{1:b-1} \mid \hat{\omega}'_k)$  in matrix  $\hat{P}'_{1:b-1}$  for iteration  $b-1$ . The shape of  $\hat{P}_{1:b-1}$  and  $\hat{P}'_{1:b-1}$  is  $K \times C^{b-1}$ .  $p(\hat{y}_b \mid \hat{\omega}_k)$  can be stored in  $\hat{P}_b$  and  $p(\hat{y}_b \mid \hat{\omega}'_k)$  in  $\hat{P}'_b$ . The

shape of  $\hat{P}_b$  and  $\hat{P}'_b$  is  $K \times C$ . Then, we compute probability of  $p(\hat{y}_{1:b})$  as follows:

$$\begin{aligned}
p(\hat{y}_{1:b}) &= \frac{1}{2K} \sum_{k=1}^K p(\hat{y}_{1:b} | \hat{\omega}_k) + p(\hat{y}_{1:b} | \hat{\omega}'_k) \\
&= \frac{1}{2K} \sum_{k=1}^K p(\hat{y}_{1:b-1} | \hat{\omega}_k) p(\hat{y}_b | \hat{\omega}_k) + p(\hat{y}_{1:b-1} | \hat{\omega}'_k) p(\hat{y}_b | \hat{\omega}'_k) \\
&= \frac{1}{2K} (\hat{P}_{1:b-1}^\top \hat{P}_b + \hat{P}'_{1:b-1} \hat{P}'_b).
\end{aligned}$$

The  $\hat{P}_{1:b-1}^\top \hat{P}_b$  and  $\hat{P}'_{1:b-1} \hat{P}'_b$  can be flattened to shape  $1 \times C^b$  after matrix multiplication. We store  $\max_{\hat{y}_{1:b-1}} p(\hat{y}_{1:b-1} | \hat{\omega}_k)$  in a matrix  $\hat{A}_{1:b-1}$  and  $\max_{\hat{y}'_{1:b-1}} p(\hat{y}'_{1:b-1} | \hat{\omega}'_k)$  in a matrix  $\hat{A}'_{1:b-1}$ . The shape of  $\hat{A}_{1:b-1}$  and  $\hat{A}'_{1:b-1}$  is  $K \times 1$ . We can compute  $\lambda_{\hat{\omega}, \hat{y}_{1:b}}$  inside edge weight discount expression by

$$\begin{aligned}
\hat{A}_{1:b} &= \hat{A}_{1:b-1} \odot \max_{\hat{y}_b} \hat{P}_b; \\
p(\hat{y}_{1:b} | \hat{\omega}_k) &= p(\hat{y}_{1:b-1} | \hat{\omega}_k) p(\hat{y}_b | \hat{\omega}_k) = \hat{P}_{1:b-1} \otimes \hat{P}_b; \\
\lambda_{\hat{\omega}, \hat{y}_{1:b}} &= \frac{p(\hat{y}_{1:b} | \hat{\omega}_k)}{\max_{\hat{y}_{1:b}} p(\hat{y}_{1:b} | \hat{\omega}_k)} = \frac{\hat{P}_{1:b-1} \otimes \hat{P}_b}{\hat{A}_{1:b}}.
\end{aligned}$$

$\odot$  is element-wise matrix multiplication and  $\otimes$  is the outer-product operator along the first dimension. After the outer product operation, we can reshape the matrix by flattening all the dimensions after 1st dimension to maintain consistency. Similarly, we can compute  $\hat{A}'_{1:b}$ ,  $p(\hat{y}_{1:b} | \hat{\omega}'_k)$  and  $\lambda_{\hat{\omega}', \hat{y}_{1:b}}$  with matrix operations. The indicator function  $\mathbb{1}_{d_H(\hat{\omega}_k, \hat{\omega}'_k) > \tau}$  can be stored in a matrix with shape  $K \times 1$ . The acquisition function can be computed with all matrix operations as follows:

$$\begin{aligned}
& \Delta_{\text{Batch-BALANCE}}(x_{1:b} \mid \mathcal{D}_{\text{train}}) \\
&= \mathbb{E}_{p(\omega \mid \mathcal{D}_{\text{train}})} \mathbb{E}_{p(y_{1:b} \mid \omega)} \left[ \mathbb{E}_{\omega, \omega' \sim p(\omega \mid \mathcal{D}_{\text{train}})} \mathbb{1}_{d_{\text{H}}(\omega, \omega') > \tau} \left( 1 - \lambda_{\omega, y_{1:b}} \lambda_{\omega', y_{1:b}} \right) \right] \\
&\approx \sum_{\hat{y}_{1:b}} \left( \frac{1}{2K} \sum_{k=1}^K p(\hat{y}_{1:b} \mid \hat{\omega}_k) + p(\hat{y}_{1:b} \mid \hat{\omega}'_k) \right) \left[ \frac{1}{K} \sum_{k=1}^K \mathbb{1}_{d_{\text{H}}(\hat{\omega}_k, \hat{\omega}'_k) > \tau} \left( 1 - \lambda_{\hat{\omega}_k, \hat{y}_{1:b}} \lambda_{\hat{\omega}'_k, \hat{y}_{1:b}} \right) \right] \\
&= \left( \frac{1}{K} \mathbb{1}_{D(\hat{\omega}_k, \hat{\omega}'_k) > \tau} \right)^\top \left( 1 - \frac{\hat{P}_{1:b-1} \otimes \hat{P}_b}{\hat{A}_{1:b}} \odot \frac{\hat{P}'_{1:b-1} \otimes \hat{P}'_b}{\hat{A}'_{1:b}} \right) \left[ \frac{1}{2K} (\hat{P}_{1:b-1}^\top \hat{P}_b + \hat{P}'_{1:b-1}^\top \hat{P}'_b) \right]^\top.
\end{aligned}$$

---

**Algorithm 4** BALANCE-Clustering

---

- 1: **input:**  $\mathcal{C} \subset \mathcal{D}_{\text{pool}}$ ,  $\bar{\mathcal{D}}_{\text{pool}}$ ,  $\{\hat{\omega}_k, \hat{\omega}'_k\}_{k=1}^K$ , threshold  $\tau$ , coldness parameter  $\beta$ , and cluster number  $B$
  - 2: sample initial centroids  $\mathcal{O} = \{\mu_j\}_{j=1}^B \subset \mathcal{C}$  with  $p(x) \sim \Delta_{\text{BALANCE}}(x)^\beta$
  - 3: **while**  $\mathcal{O}$  not converged **do**
  - 4:   **for all**  $x \in \mathcal{C}$  **do**
  - 5:      $a_x \leftarrow \arg \max_j I_{\Delta_{\text{BALANCE}}}(x, \mu_j)$
  - 6:    $\mathcal{S}_j \leftarrow \{x \in \mathcal{C} : a_x = j\}$
  - 7:   **for all**  $j \in [B]$  **do**
  - 8:      $\mu_j \leftarrow \arg \max_{y \in \mathcal{S}_j} \sum_{x \in \mathcal{S}_j} I_{\Delta_{\text{BALANCE}}}(x, y)$
  - 9: **output:**  $\mathcal{S}_{1:B}, \mu_{1:B}$
- 

### 6.3.3 Stochastic Selection with Power Sampling and BALANCE-Clustering

A simple approach to apply our new acquisition function to a large batch is stochastic batch selection [Kirsch et al., 2021a], where we randomly select a batch with power distribution  $p(x) \sim \Delta_{\text{BALANCE}}(x)^\beta$ . We call this algorithm PowerBALANCE.

Next, we sought to further improve PowerBALANCE through a novel acquisition-function-driven clustering procedure. Inspired by Kothawade et al. [2021], we define a novel *infor-*



information measure  $I_{\Delta_{\text{BALANCE}}}(x, y)$  for any two data samples  $x$  and  $y$  based on our acquisition function:

$$I_{\Delta_{\text{BALANCE}}}(x, y) = \Delta_{\text{BALANCE}}(x) + \Delta_{\text{BALANCE}}(y) - \Delta_{\text{BALANCE}}(\{x, y\}) \quad (6.6)$$

Intuitively,  $I_{\Delta_{\text{BALANCE}}}(x, y)$  captures the amount of overlap between  $x$  and  $y$  w.r.t.  $\Delta_{\text{BALANCE}}$ . Therefore, it is natural to use it as a similarity measure for clustering and use the cluster centroids as candidate queries. The BALANCE-Clustering algorithm is illustrated in algorithm 4.

Concretely, we first sample a subset  $\mathcal{C} \subset \mathcal{D}_{\text{pool}}$  with  $p(x) \sim \Delta_{\text{BALANCE}}(x)^\beta$  similar to [Kirsch et al., 2021a]. The BALANCE-Clustering then runs an Lloyd’s algorithm (with a non-Euclidean metric) to find  $B$  cluster centroids (see Line 3-8 in algorithm 4): it takes the subset  $\mathcal{C}$ ,  $\{\hat{\omega}_k, \hat{\omega}'_k\}_{k=1}^K$ , threshold  $\tau$ , coldness parameter  $\beta$ , and cluster number  $B$  as input. It first samples initial centroids  $\mathcal{O}$  with  $p(x) \sim \Delta_{\text{BALANCE}}(x)^\beta$ . Then, it iterates the process of adjusting the clusters and centroids until convergence and outputs  $B$  cluster centroids as candidate queries.

## 6.4 Experiments

In this section, we sought to show the efficacy of Batch-BALANCE on several diverse datasets, under both small batch setting and large batch setting.

### 6.4.1 Datasets

In the main paper, we consider four datasets (i.e. MNIST [LeCun et al., 1998], Repeated-MNIST [Kirsch et al., 2019], Fashion-MNIST [Xiao et al., 2017] and EMNIST [Cohen et al., 2017]) as benchmarks for the small-batch setting, and two datasets (i.e. SVHN [Netzer et al., 2011], CIFAR [Krizhevsky et al., 2009]) as benchmarks for the large-batch setting.

The reason for making the splits is that for the more challenging classification tasks on SVHN and CIFAR-10, the performance improvement for all baseline algorithms from a small batch (e.g., with batch size  $< 50$ ) is hardly visible. We split each dataset into unlabeled AL pool  $\mathcal{D}_{\text{pool}}$ , initial training dataset  $\mathcal{D}_{\text{train}}$ , validation dataset  $\mathcal{D}_{\text{val}}$ , test dataset  $\mathcal{D}_{\text{test}}$ , and unlabeled dataset  $\bar{\mathcal{D}}_{\text{pool}}$ .  $\bar{\mathcal{D}}_{\text{pool}}$  is only used for calculating the Hamming distance between hypotheses and is never used for training BNNs.

**MNIST.** We randomly split the MNIST training dataset into  $\mathcal{D}_{\text{val}}$  with 10,000 samples,  $\bar{\mathcal{D}}_{\text{pool}}$  with 10,000 samples, and  $\mathcal{D}_{\text{pool}}$  with the rest. The initial training dataset contains 20 samples with 2 samples in each class chosen from the AL pool. The BNN model architecture is similar to Kirsch et al. [2019]. It consists of two blocks of [convolution, dropout, max-pooling, relu] followed by a two-layer MLP that a two-layer MLP and one dropout between the two layers. The dropout probability is 0.5 in the dropout layers.

**Repeated-MNIST.** Kirsch et al. [2019] show that applying BALD to a dataset that contains many (near) replicated data points leads to poor performance. We again randomly split the MNIST training dataset similar to the settings used on the MNIST dataset. We replicate all the samples in the AL pool two times and add isotropic Gaussian noise with a standard deviation of 0.1 after normalizing the dataset. The BNN architecture is the same as the one used on the MNIST dataset.

**EMNIST.** We further consider the EMNIST dataset under 3 different settings: EMNIST-Balanced, EMNIST-ByClass, and EMNIST-ByMerge. The EMNIST-Balanced contains 47 classes with balanced digits and letters. EMNIST-ByMerge includes digits and letters for a total of 47 unbalanced classes. EMNIST-ByClass represents the most useful organization for classification as it contains the segmented digits and characters for 62 classes comprising [0-9],[a-z], and [A-Z]. We randomly split the training set into  $\mathcal{D}_{\text{val}}$  with 18,800 images,  $\bar{\mathcal{D}}_{\text{pool}}$

with 18,800 images, and  $\mathcal{D}_{\text{pool}}$  with the rest of the samples. Similar to Kirsch et al. [2019], we do not use an initial dataset and instead perform the initial acquisition step with the randomly initialized model. The model architecture contains three blocks of [convolution, dropout, max-pooling, relu], with 32, 64, and 128 3x3 convolution filters and 2x2 max pooling. We add a two-layer MLP following the three blocks. 4 dropout layers in total are in each block and MLP with a dropout probability of 0.5.

**Fashion-MNIST** Fashion-MNIST is a dataset of Zalando’s article images that consists of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes. We randomly split the Fashion-MNIST training dataset into  $\mathcal{D}_{\text{val}}$  with 10,000 samples,  $\bar{\mathcal{D}}_{\text{pool}}$  with 10,000 samples, and  $\mathcal{D}_{\text{pool}}$  with the rest of the samples. We obtain the initial training dataset that contains 20 samples with 2 samples in each class randomly chosen from the AL pool. The model architecture is similar to the one used on the EMNIST dataset with 10 units in the last MLP.

**SVHN** We randomly select initial training dataset with 5,000 samples,  $\bar{\mathcal{D}}_{\text{pool}}$  with 2,000 samples, and validation dataset  $\mathcal{D}_{\text{val}}$  with 5,000 samples. Similarly for CIFAR-10 dataset,

**CIFAR** we random select initial training dataset with 5,000 samples,  $\bar{\mathcal{D}}_{\text{pool}}$  with 5,000 samples, and validation dataset  $\mathcal{D}_{\text{val}}$  with 5,000 samples.

#### 6.4.2 Experimental Setup

**BNN models** At each AL iteration, we sample BNN posteriors given the acquired training dataset and select samples from  $\mathcal{D}_{\text{pool}}$  to query labels according to the acquisition function of a chosen algorithm. To avoid overfitting, we train the BNNs with MC dropout at each iteration with early stopping. for MNIST, Repeated-MNIST, EMNIST, and FashionMNIST,

we terminate the training of BNNs with a patience of 3 epochs. For SVHN and CIFAR-10, we terminate the training of BNNs with a patience of 20 epochs. The BNN with the highest validation accuracy is picked and used to calculate the acquisition functions. Additionally, we use weighted cross-entropy loss for training the BNN to mitigate the bias introduced by imbalanced training data. The BNN models are reinitialized in each AL iteration similar to Gal et al. [2017], Kirsch et al. [2019]. It decorrelates subsequent acquisitions as the final model performance is dependent on a particular initialization. We use Adam optimizer [Kingma and Ba, 2017] for all the models in the experiments.

For cSG-MCMC, we use ResNet-18 [He et al., 2016] and run 400 epochs in each AL iteration. We set the number of cycles to 8 and the initial step size to 0.5. 3 samples are collected in each cycle.

**Acquisition criterion for Batch-BALANCE under different batch sizes** For small AL batch with  $B < 50$ , Batch-BALANCE takes the greedy selection approach. For large AL batch with  $B \geq 50$ , BALANCE takes the clustering approach described in §6.3.3. In the small batch-mode setting, if  $b < 4$ , Batch-BALANCE enumerates all  $y_{1:b}$  configurations to compute the acquisition function  $\Delta_{(\text{Batch-})\text{BALANCE}}$  according to Eq. (6.4); otherwise, it uses  $M = 10,000$  MC samples of  $y_{1:b}$  and importance sampling to estimate  $\Delta_{\text{Batch-BALANCE}}$  according to Eq. (6.5). All our results report the median of 6 trials, with lower and upper quartiles.

**Baselines** For the small-batch setting, we compare Batch-BALANCE with Random, Variation Ratio [Freeman and Freeman, 1965], Mean STD [Kendall et al., 2015] and BatchBALD. To the best of the authors’ knowledge, Batch-BALD still achieves state-of-the-art performance for deep Bayesian AL with small batches. For large-batch setting, it is no longer feasible to run BatchBALD [Citovsky et al., 2021]; we consider other baseline models both in Bayesian setting, e.g., PowerBALD, and Non-Bayesian setting, e.g., CoreSet and BADGE.

AL algorithms	Complexity
Mean STD	$\mathcal{O}( \mathcal{D}_{\text{pool}} (CK + \log B))$
Variation Ratio	$\mathcal{O}( \mathcal{D}_{\text{pool}} (CK + \log B))$
PowerBALD	$\mathcal{O}( \mathcal{D}_{\text{pool}} (CK + \log B))$
BatchBALD	$\mathcal{O}( \mathcal{D}_{\text{pool}} BMK)$
CoreSet (2-approx)	$\mathcal{O}( \mathcal{D}_{\text{pool}} HB)$
BADGE	$\mathcal{O}( \mathcal{D}_{\text{pool}} HCB^2)$
PowerBALANCE	$\mathcal{O}( \mathcal{D}_{\text{pool}} (C \cdot 2K + \log B))$
Batch-BALANCE (GreedySelection)	$\mathcal{O}( \mathcal{D}_{\text{pool}} BM \cdot 2K)$
Batch-BALANCE (BALANCE-Clustering)	$\mathcal{O}( \mathcal{D}_{\text{pool}} C \cdot 2K +  \mathcal{C} ^2(C^2 \cdot 2K + T))$

Table 6.1: Computational complexity of AL algorithms.

### 6.4.3 Computational Complexity Analysis

Table 6.1 shows the computational complexity of the batch-mode AL algorithms evaluated in this paper. Here,  $C$  denotes the number of classes,  $B$  denotes the acquisition size,  $K$  is the pair number of posterior samples and  $M$  is the sample number for  $y_{1:b}$  configurations. We assume the number of the hidden units is  $H$ .  $T$  is # iterations for BALANCE-Clustering to converge and is usually less than 5. In Figure. 6.3 we plot the computation time for a single batch (in seconds) by different algorithms. As the batch size increases, variants of Batch-BALANCE (including Batch-BALANCE and PowerBALANCE as its special case) both outperform CoreSet in run time. In later subsections, we will demonstrate that this gain in computational efficiency does not come at a cost of performance.

### 6.4.4 Batch-mode Deep Bayesian AL with Small Batch Size

We compare 5 different models with acquisition sizes  $B = 1$ ,  $B = 3$ , and  $B = 10$  on the MNIST dataset.  $K = 100$  for all the methods. The threshold  $\tau$  for Batch-BALANCE is annealed by setting  $\tau$  to  $\varepsilon/2$  in each AL loop. Note that when  $B = 3$ , we can compute the acquisition function with all  $y_{1:b}$  configurations for  $b = 1, 2, 3$ . When  $b \geq 4$ , we approximate

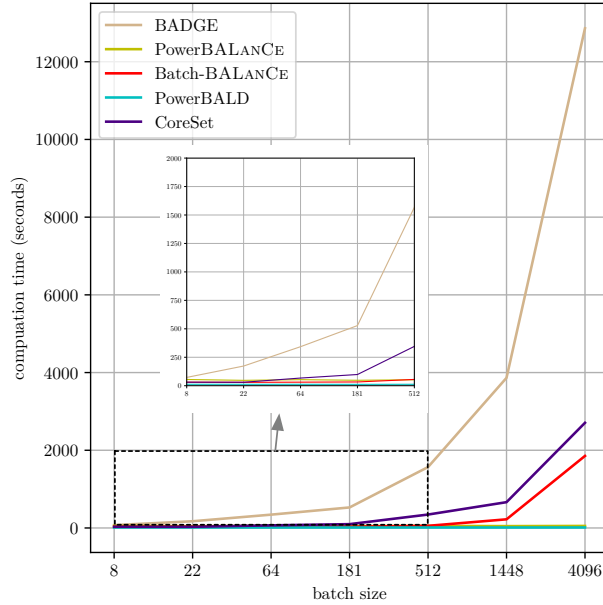


Figure 6.3: Run time vs. batch size.

the acquisition function with importance sampling. Figure. 6.4 (a)-(c) show that Batch-BALANCE are consistently better than other baseline methods for the MNIST dataset.

We then compare Batch-BALANCE with other baseline methods on three datasets with balanced classes—Repeated-MNIST, Fashion-MNIST, and EMNIST-Balanced. The acquisition size  $B$  for Repeated-MNIST and Fashion-MNIST is 10 and is 5 for the EMNIST-Balanced dataset. The threshold  $\tau$  of Batch-BALANCE is annealed by setting  $\tau = \varepsilon/4^4$ . The learning curves of accuracy are shown in Figure. 6.4 (d)-(f). For the Repeated-MNIST dataset, BALD performs poorly and is worse than random selection. BatchBALD is able to cope with the replication after a certain number of AL loops, which is aligned with the result shown in Kirsch et al. [2019]. Batch-BALANCE is able to beat all the other methods on this dataset.

For the Fashion-MNIST dataset, Batch-BALANCE outperforms random selection but the other methods fail. For the EMNIST dataset, Batch-BALANCE is slightly better than BatchBALD.

---

4. Empirically we find that  $\tau \in [\varepsilon/8, \varepsilon/2]$  works generally well for all datasets.

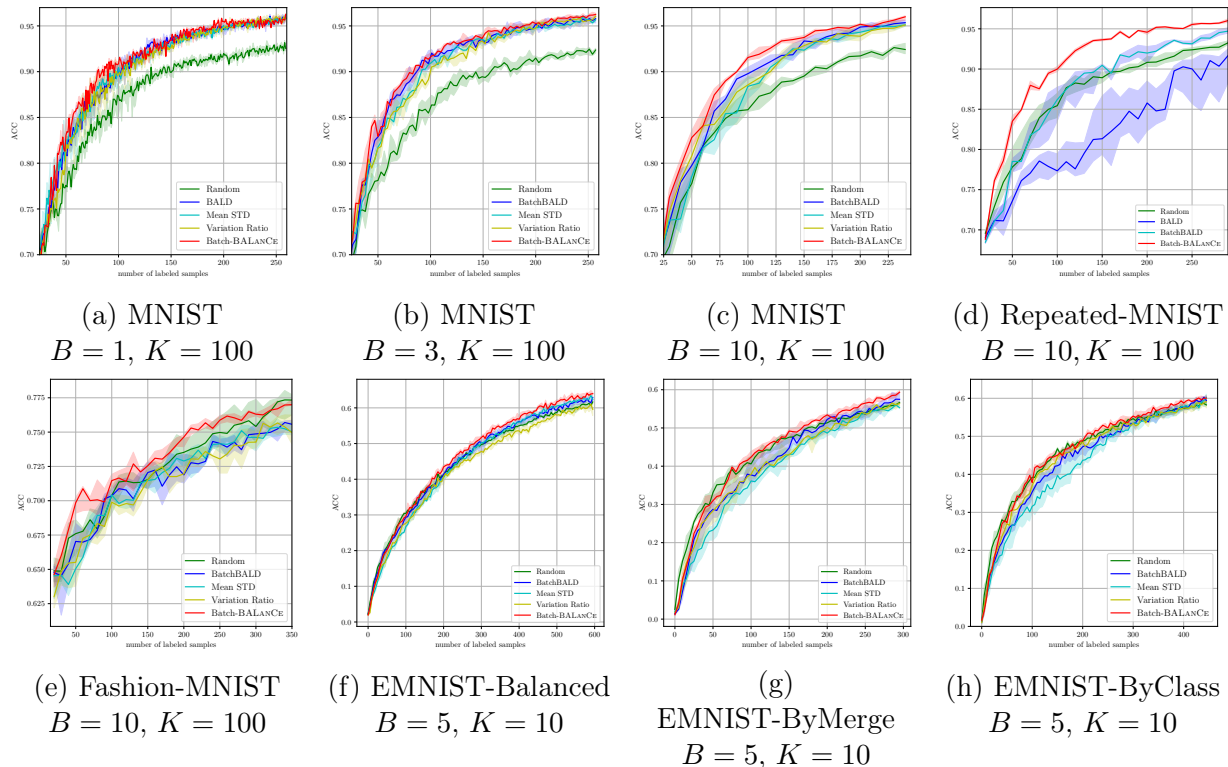


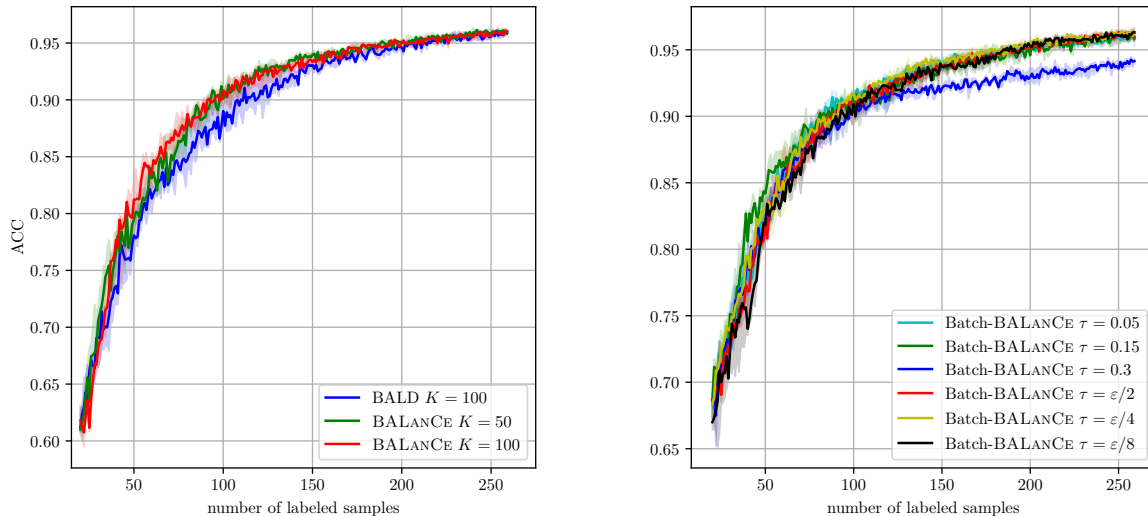
Figure 6.4: Experimental results on MNIST, Repeated-MNIST, Fashion-MNIST, EMNIST-Balanced, EMNIST-ByClass, and EMNIST-ByMerge datasets in the small-batch regime. For all plots, the  $y$ -axis represents accuracy and  $x$ -axis represents the number of queried examples.

We further compare different algorithms with two unbalanced datasets: EMNIST-ByMerge and EMNIST-ByClass. The  $\tau$  for Batch-BALANCE is set  $\varepsilon/4$  in each AL loop.  $B = 5$  and  $K = 10$  for all the methods. As pointed out by Kirsch et al. [2019], BatchBALD performs poorly in unbalanced dataset settings. BALANCE and Batch-BALANCE can cope with the unbalanced data settings. The result is shown in Figure. 6.4 (g) and (h).

#### 6.4.5 Effect of Different Choices of Hyperparameters

We compare BALD and BALANCE with batch size  $B = 1$  and different  $K$ 's on an imbalanced MNIST dataset which is created by removing a random portion of images for each class in the training dataset. Figure. 6.5 (a) shows that BALANCE performs the best with a large

margin to the curve of BALD. Note that BALANCE with  $K = 50$  is also better than BALD with  $K = 100$ .



(a) ACC vs. # samples for different  $K$ 's.

(b) ACC vs. # samples for different  $\tau$ 's.

Figure 6.5: Learning curves of different  $K$  and  $\tau$  for BALANCE.

We also study the influence of  $\tau$  for BALANCE on the MNIST dataset. Denote the validation error rate of the BNN model by  $\epsilon$ . BALANCE with fixed  $\tau = 0.05, 0.15, 0.3$  and annealing  $\tau = \epsilon/2, \epsilon/4, \epsilon/8$  are run on MNIST dataset and the learning curves are shown in Figure. 6.5 (b). The BALANCE is robust to  $\tau$ . However, when  $\tau$  is set to 0.3 and the test accuracy gets around 0.88, the accuracy improvement becomes slow. The reason for this slow improvement is that the threshold  $\tau$  is too large and all the pairs of posterior samples are treated as in the same equivalence class and the acquisition functions for all the samples in the AL pool are zeros. In other words, the BALANCE degrades to random selection when  $\tau$  is too large.

We further pick a data point from this imbalanced MNIST dataset and gradually increase the posterior sample number  $K$  to estimate the acquisition function value  $\Delta_{\text{BALANCE}}$  for this data point. For each posterior sample number  $K$ , we estimate the acquisition function



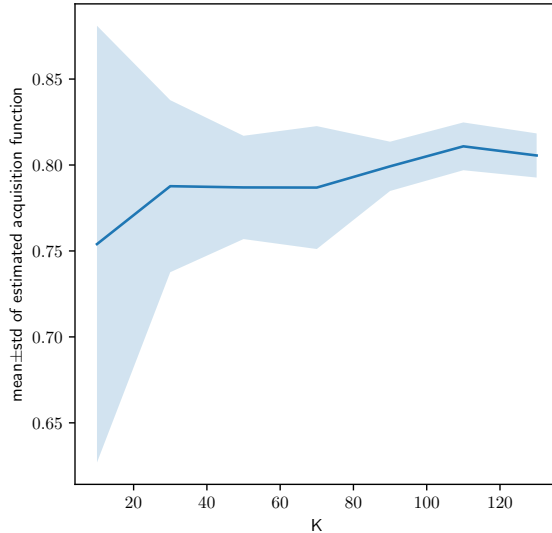


Figure 6.6: Estimated acquisition function values  $\Delta_{\text{BALANCE}}$  of BALANCE vs. posterior sample number  $K$

$\Delta_{\text{BALANCE}}$  10 times with 10 sets of posterior sample pairs. The mean and std for this  $K$  are calculated and shown in Figure. 6.6.

#### 6.4.6 Experiments on Tabular Datasets

We compare different AL algorithms on tabular datasets including Human Activity Recognition Using Smartphones Data Set [Anguita et al., 2013] (HAR), Gas Sensor Array Drift [Vergara et al., 2012] (DRIFT), and Dry Bean Dataset [Koklu and Ozkan, 2020], as well as a more difficult dataset CINIC-10 [Darlow et al., 2018].

**HAR, DRIFT and Dry Bean Dataset** We run 6 AL trials for each dataset and algorithm. In each iteration, the BNNs are trained with a learning rate of 0.01 and patience equal to 3 epochs. The BNNs all contain three-layer MLP with ReLU activation and dropout layers in between. The datasets are all split into a starting training set, validation set, testing set, and AL pool. The AL pool is also used as  $\bar{\mathcal{D}}_{\text{pool}}$ . The  $\tau$  for Batch-BALANCE is set  $\varepsilon/4$

in each AL loop. See Table 6.2 for more experiment details of these 3 datasets.

dataset	val set size	test set size	hidden unit #	sample # per epoch	K	B
HAR	2K	2,947	(64,64)	4,096	20	10
DRIFT	2K	2K	(32,32)	4,096	20	10
Dry Bean	2K	2K	(8,8)	8,192	20	10

Table 6.2: Experment details for HAR, DRIFT and Dry Bean Dataset

The learning curves of all 5 algorithms on these 3 tabular datasets are shown in Figure. 6.7. Batch-BALANCE outperforms all the other algorithms for these 3 datasets. For the HAR dataset, both Batch-BALANCE and BatchBALD work better than random selection. In Figure. 6.7 (b) and (c), Mean STD, Variation Ratio, and BatchBALD perform worse than random selection. We find a similar effect for some other imbalanced datasets.

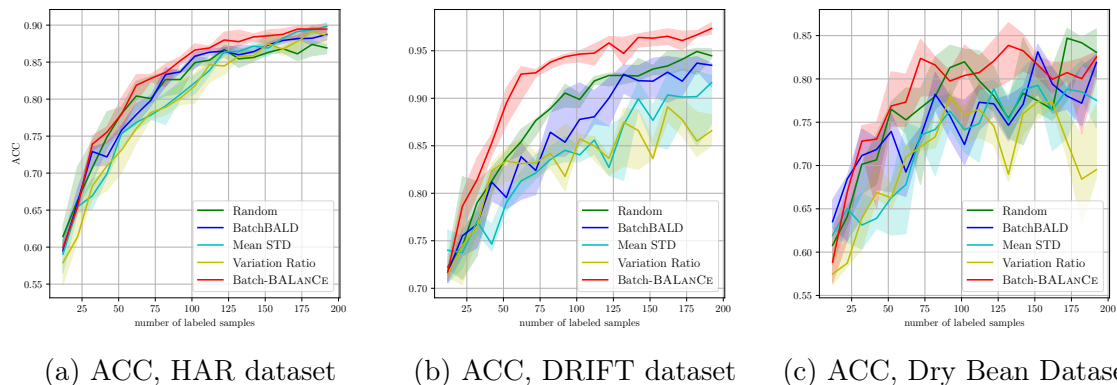


Figure 6.7: Experimental results on 3 tabular datasets. For all plots, the  $y$ -axis represents accuracy and  $x$ -axis represents the number of queried examples.

**CINIC-10** CINIC-10 is a large dataset with 270K images from two sources: CIFAR-10 [Krizhevsky et al., 2009] and ImageNet [Rasmus et al., 2015]. The training set is split into an AL pool with 120K samples, 40K  $\bar{D}_{\text{pool}}$  samples, 20K validation samples, and 200 starting training samples with 20 samples in each class. We use VGG-11 as the BNN. The number of sampled MC dropout pairs is 50 and the acquisition size is 10. We run 6 trials for this experiment. The learning curves of 5 algorithms are shown in Figure. 6.8. We can see from

Figure. 6.8 that Batch-BALANCE performs better than all the other algorithms by a large margin in this setting.

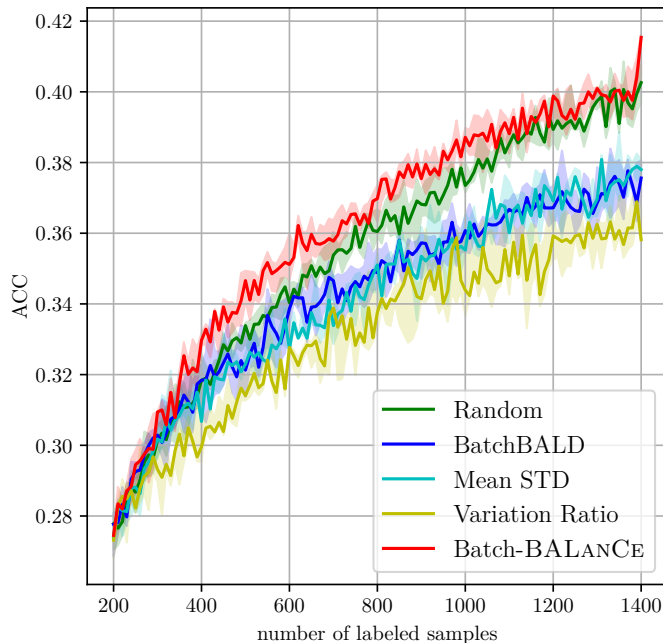
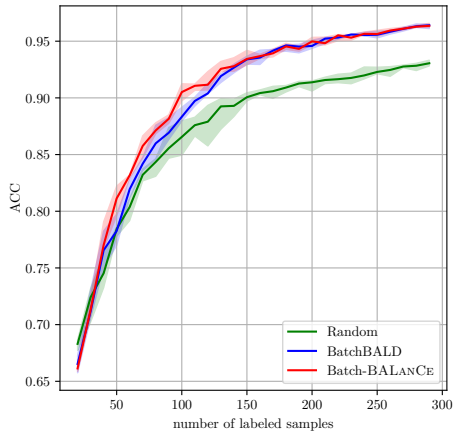


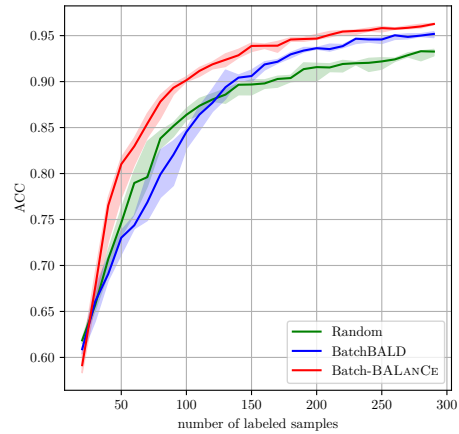
Figure 6.8: ACC vs. # samples on the CINIC-10 dataset.

**Repeated-MNIST with different amounts of repetitions** In order to show the effect of redundant data points on BathBALD and Batch-BALANCE, we ran experiments on Repeated-MNIST with an increasing number of repetitions. The learning curves of accuracy for Repeated-MNIST with different repetition numbers can be seen in Figure. 6.9. A detailed model accuracy on the test dataset when the acquired training dataset size is 130 is shown in Table 6.3. Even though Batch-BALANCE can improve data efficiency [Kirsch et al., 2019], there are still large gaps between the learning curves of Batch-BALD and Batch-BALANCE and the gaps become larger when the number of repetitions increases.

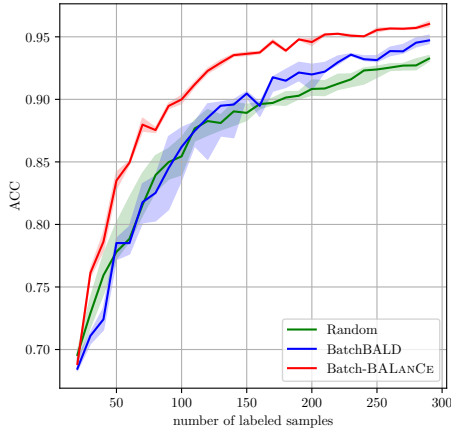
In order to compare our algorithms with other AL algorithms in this small batch size regime, we further run PowerBALANCE, PowerBALD, BADGE, and CoreSet on the Repeated-



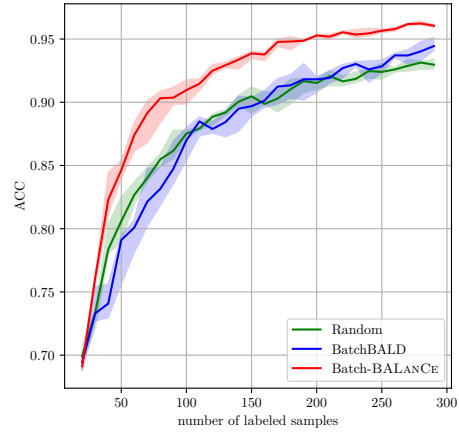
(a) repeat 0 time



(b) repeat 1 times



(c) repeat 2 times



(d) repeat 3 times

Figure 6.9: Performance of Random selection, BatchBALD, and Batch-BALANCE on Repeated-MNIST for an increasing number of repetitions. For all plots, the  $y$ -axis represents accuracy and the  $x$ -axis represents the number of queried examples. We can see that BatchBALD also performs worse as the number of repetitions is increased. Batch-BALANCE outperforms BatchBALD with large margins and remains similar performance across different numbers of repetitions.

MNIST with repeat number 3. As shown in Figure. 6.10, Batch-BALANCE achieves the best performance. Note that both PowerBALD and PowerBALANCE are efficient in selecting AL batch and show similar performance compared to the BADGE algorithm.

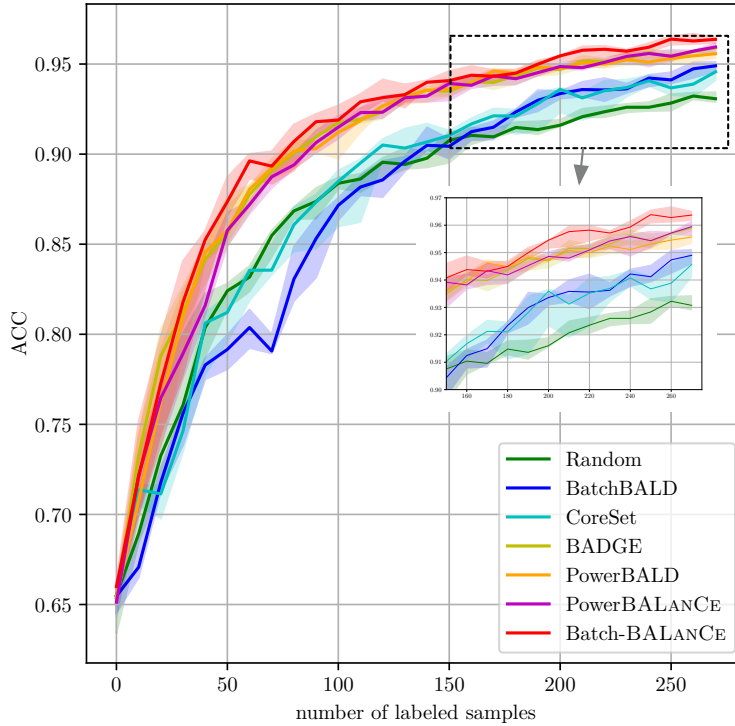


Figure 6.10: ACC vs. # samples on RepeatedMNIST dataset with repeat number 3.

**CIFAR-100** For CIFAR-100, we use 100 fine-grained labels. The dataset is split into an initial training dataset with 5,000 samples,  $\bar{\mathcal{D}}_{\text{pool}}$  with 5,000 samples, and a validation dataset  $\mathcal{D}_{\text{val}}$  with 5,000 samples. The experiment is conducted with batch size  $B = 5,000$  and a budget of 25,000. The cSG-MCMC is used for BNN with epoch number 200, initial step size 0.5, and cycle number 4. We can see in Figure. 6.11 that both PowerBALANCE and Batch-BALANCE perform well in this dataset.

#### 6.4.7 Additional Evaluation Metrics

Besides accuracy, we compared macro-average AUC, macro-average F1, and NLL for 5 different methods on EMNIST-Balanced and EMNIST-ByMerge datasets in Figure. 6.12. The acquisition size for all the AL algorithms is 5. Batch-BALANCE is annealed by setting  $\tau = \varepsilon/4$ . A macro-average AUC computes the AUC independently for each class and then

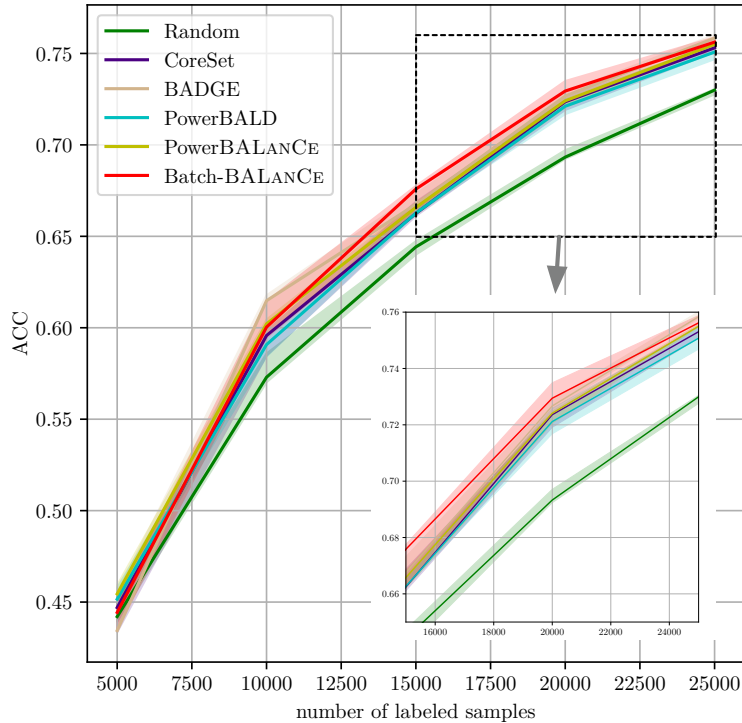


Figure 6.11: ACC vs. # samples, cSG-MCMC, CIFAR-100

takes the average. Both macro-average AUC and macro-average F1 take class imbalance into account. As shown in Figure. 6.12, Batch-BALANCE attains better data efficiency compared with baseline models on both balanced and imbalanced datasets.

We also evaluated the negative log-likelihood (NLL) for different AL algorithms. NLL is a popular metric for evaluating predictive uncertainty [Quinonero-Candela et al., 2005]. As shown in Figure. 6.12, Batch-BALANCE maintains a better or comparable quality of predictive uncertainty over test data.

#### 6.4.8 BALANCE via Explicit Partitioning over the Hypothesis Posterior

##### Samples

Another way of estimating the acquisition function is to construct the equivalence classes explicitly first (e.g. by partitioning the hypothesis spaces into  $k$  Voronoi cells via max-

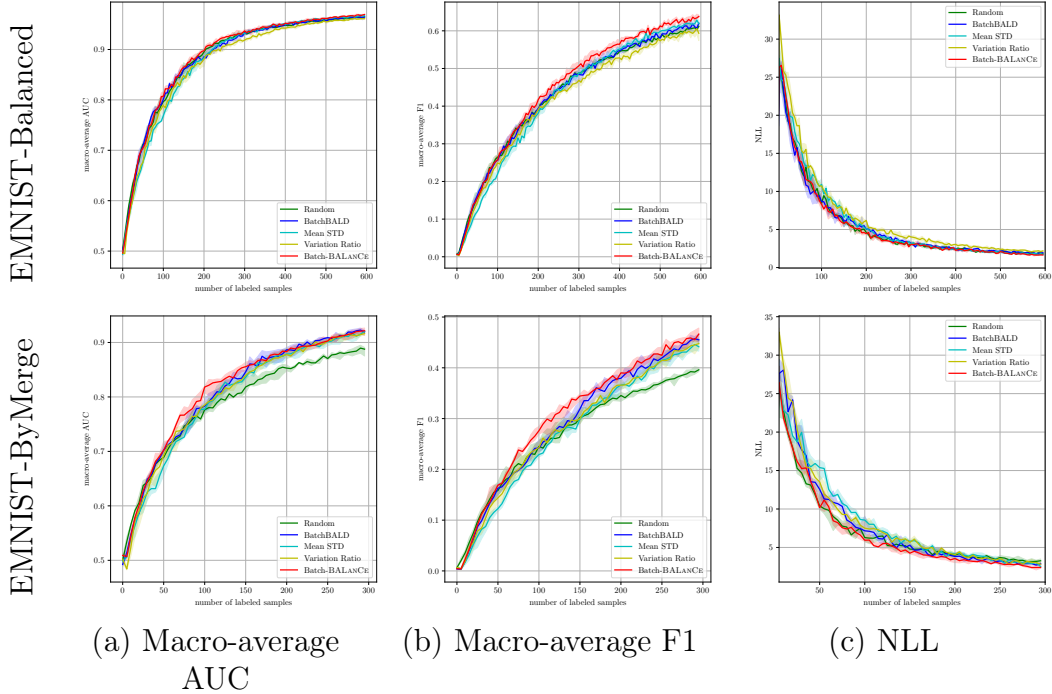


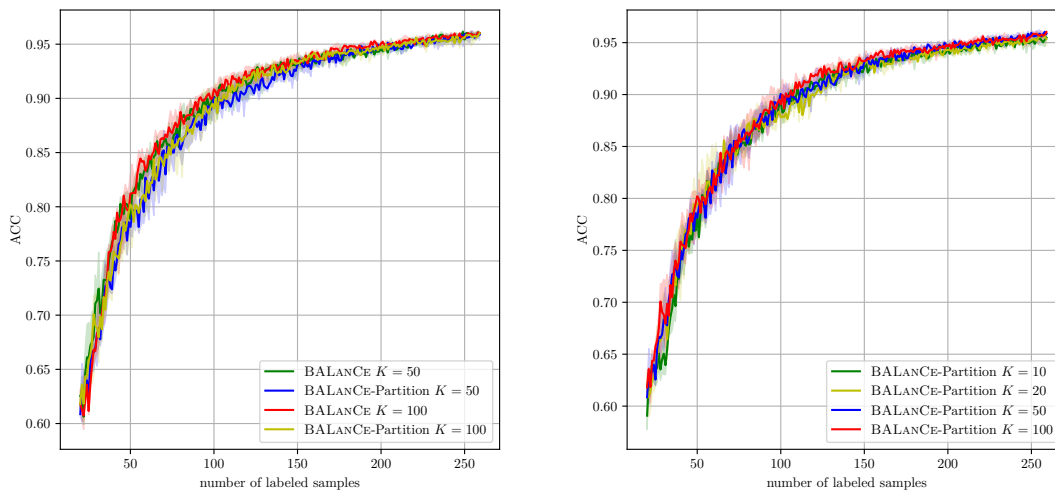
Figure 6.12: Compare different metrics for EMNIST-Balanced and EMNIST-Bymerge

diameter clustering and calculate the weight discounts of edges that connect different equivalence classes. Intuitively, explicitly constructing equivalence classes may introduce unnecessary edges as two closeby hypotheses can be partitioned into different equivalence classes; therefore leading to an overestimate of the edge weight discounted. We call this algorithm BALANCE-Partition.

In order to compare with BALANCE and Batch-BALANCE, we sampled  $K$  pairs of MC dropouts to estimate the acquisition function of BALANCE-Partition. All the representations of  $2K$  MC dropouts on  $\bar{\mathcal{D}}_{\text{pool}}$  are generated. We run FFT [Gonzalez, 1985] with Hamming distances and threshold  $\tau$  on these representations to get approximated ECs. Each data point has at most  $\tau$  Hamming distance to the corresponding cluster center. FFT is a 2-approx algorithm and the optimal solution with the same cluster number has cluster diameter  $\geq \frac{\tau}{2}$ . After equivalence classes are returned, BALANCE-Partition calculates the edges discounts of all edges that connect different equivalence classes and estimates the acquisition

function values of each data sample in the AL pool.

Although a faster method that utilizes complete homogeneous symmetric polynomials [Javdani et al., 2014] can be implemented to estimate the acquisition function values for BALANCE-Partition, experiments in Figure. 6.13 show that BALANCE-Partition can not achieve better performance than BALANCE and increasing the MC dropout number does not improve performance significantly.



(a) Compare BALANCE-Partition with BALANCE

(b) BALANCE-Partition with different  $K$

Figure 6.13: ACC vs. # samples for BALANCE-Partition and BALANCE.

Method	repeat 1 time	repeat 2 times	repeat 3 times	repeat 4 times
Random	$0.887 \pm 0.017$	$0.883 \pm 0.012$	$0.881 \pm 0.013$	$0.895 \pm 0.009$
BatchBALD	$0.917 \pm 0.005$	$0.892 \pm 0.023$	$0.883 \pm 0.025$	$0.881 \pm 0.014$
Batch-BALANCE	$0.926 \pm 0.008$	$0.923 \pm 0.008$	$0.929 \pm 0.004$	$0.927 \pm 0.010$

Table 6.3: Mean $\pm$ STD of test accuracies when acquired training set size is 130

#### 6.4.9 Batch-mode Deep Bayesian AL with Large Batch Size

**Batch-BALANCE with MC dropout** We test different AL models on two larger datasets with larger batch sizes. The acquisition batch size  $B$  is set to 1,000 and  $\tau = \varepsilon/8$ . We use



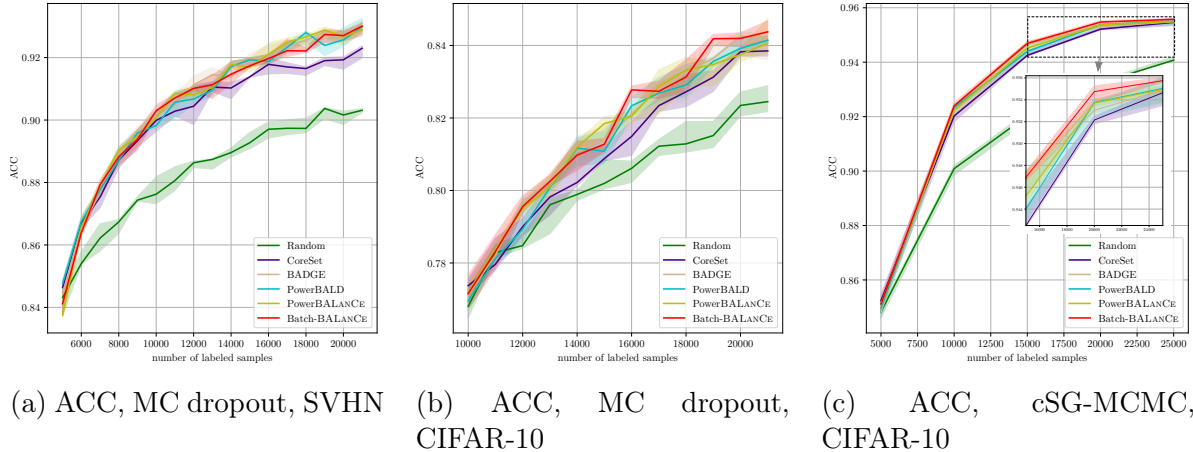


Figure 6.14: Performance on SVHN and CIFAR-10 datasets in the large-batch regime.

VGG-11 as the BNN and train it on all the labeled data with patience equal to 20 epochs in each AL iteration. The VGG-11 is trained using SGD with a fixed learning rate of 0.001 and momentum of 0.9. The size of  $\mathcal{C}$  for Batch-BALANCE is set to  $2B$ . Similar to PowerBALD [Kirsch et al., 2021a], we also find that PowerBALANCE and BatchBALANCE are insensitive to  $\beta$  and  $\beta = 1$  works generally well. We thus set the coldness parameter  $\beta = 1$  for all algorithms.

The performance of different AL models on these two datasets is shown in Figure. 6.14 (a) and (b). PowerBALD, PowerBALANCE, BADGE, and BatchBALANCE get similar performance on SVHN dataset. For the CIFAR-10 dataset, BatchBALANCE shows compelling performance. Note that PowerBALANCE also performs well compared to other methods.

**Batch-BALANCE with cSG-MCMC** We test different AL models with cSG-MCMC on CIFAR-10. The acquisition batch size  $B$  is 5,000. The size of  $\mathcal{C}$  for Batch-BALANCE is set to  $3B$ . In order to apply the CoreSet algorithm to BNN, we use the average activations of all posterior samples’ final fully-connected layers as the representations. For BADGE, we use the label with maximum average predictive probability as the hallucinated label and use the average loss gradient of the last layer induced by the hallucinated label as the representation.

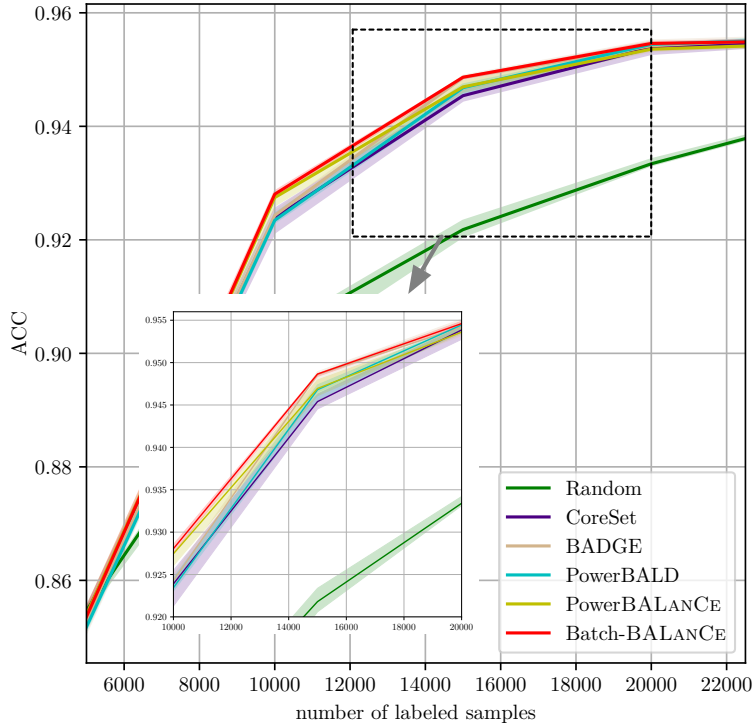


Figure 6.15: ACC vs. # samples, multi-chain cSG-MCMC, CIFAR-10

We can see from Figure. 6.14 (c) that Batch-BALANCE achieves the best performance.

#### 6.4.10 Batch-BALANCE with Multi-chain cSG-MCMC

cSG-MCMC can be improved by sampling with multiple chains [Zhang et al., 2019]. In order to evaluate different AL algorithms with this improved parallel cSG-MCMC method, we conduct experiment on the CIFAR-10 dataset with batch size  $B = 5,000$ . We sample posteriors with 3 chains. Each chain trains the model 200 epochs. The cycle number for each chain is 4 and 3 posterior samples are collected in each cycle. The result is shown in Figure. 6.15, Batch-BALANCE achieves better performance than BADGE.

## 6.5 Conclusion

We have proposed a scalable batch-mode deep Bayesian active learning framework, which leverages the hypothesis structure captured by equivalence classes without explicitly constructing them. Batch-BALANCE selects a batch of samples at each iteration, which can reduce the overhead of retraining the model and save labeling effort. By combining insights from decision-theoretic active learning and diversity sampling, the proposed algorithms achieve compelling performance efficiently on active learning benchmarks both in small batch- and large batch-mode settings. Given the promising empirical results on the standard benchmark datasets explored in this paper, we are further interested in understanding the theoretical properties of the equivalence annealing algorithm under controlled studies as future work.

# CHAPTER 7

## CONCLUSION AND OUTLOOK

### 7.1 Conclusion and Discussion

This dissertation has made several significant contributions to the field of machine learning with a focus on the analysis of histopathology images in low-data regimes. The primary goal was to develop and evaluate methods that enhance the efficiency and effectiveness of machine learning applications from caption generation and image classification to active learning strategies and representation learning. Each of these contributions not only addresses specific challenges within the field but also opens up new avenues for research and application.

The introduction of the PathCap and PathHyperbolic models represents a foundational advancement in the use of deep learning for interpreting complex medical images. The PathCap model leverages multi-scale views to generate accurate, informative captions for whole-slide histopathology images, significantly outperforming baseline models. This not only aids in standardizing clinical ontologies but also improves the accessibility and annotation quality of medical images, which is critical for both educational and diagnostic purposes in medical fields. Similarly, the PathHyperbolic model integrates hyperbolic spaces with attention mechanisms to enhance image classification tasks. This novel approach has shown superior performance by effectively highlighting discriminative structures across various scales, thus providing a more nuanced analysis than traditional models. The success of these models demonstrates the potential of advanced machine learning techniques in transforming medical image analysis, offering more precise and interpretable results that can greatly benefit clinical practices.

The development of the Batch-BALANCE algorithm underlines the effectiveness of using deep Bayesian active learning frameworks to manage the scarcity of labeled data in medical imaging fields. By innovatively applying decision-theoretic principles and combinatorial

optimization, this approach not only refines the model’s learning process but also significantly reduces the cost and effort required in data annotation. Batch-BALANCE’s capability to efficiently select informative samples from large datasets without compromising performance is a critical enhancement that promises to streamline workflows in clinical image analysis.

The exploration of representation learning using coarse-grained labels has set a new precedent in the utilization of available data. By focusing on the hierarchical relationships between different data granularities, the proposed few-shot learning algorithm efficiently predicts fine-grained labels even from limited data. This approach not only circumvents the challenge of acquiring extensive fine-grained annotations but also maximizes the predictive performance using minimal resources, showcasing the feasibility of sophisticated machine learning models in resource-constrained settings.

In conclusion, this dissertation represents a significant leap forward in applying machine learning to enhance histopathology image analysis in scenarios characterized by data scarcity. By introducing groundbreaking models like PathCap and PathHyperbolic, and by advancing active learning strategies through the Batch-BALANCE algorithm, this work has effectively pushed the boundaries of what is achievable in medical image analysis. Furthermore, the innovative use of representation learning with coarse-grained labels exemplifies a smart approach to overcoming common data limitations in medical settings. These contributions not only fulfill the dissertation’s primary goals but also establish a solid foundation for future research, offering promising pathways for both academic exploration and practical implementation in medical diagnostics and education. The techniques developed here hold the potential to significantly influence clinical practices and patient outcomes by enhancing the accuracy and efficiency of medical diagnostics through advanced machine learning.

## Future Directions

We require an efficient and precise analysis system for histopathology images. The development of such systems requires collaborative, interdisciplinary approaches that translate diverse sources of raw information into accessible scientific insight. To this end, more research can be done to expand upon a strong foundation built by our past and current research. Furthermore, some efficient strategies can be designed to use available data and make effective use of data from more recent technologies.

### *Active Data Acquisition and Subset Selection from Source Domain*

Transfer learning/broad transfer [Ilharco et al., 2022] offers great potential to adapt foundation models to specialized domains. However, the domain shift poses intertwined challenges for active data acquisition and (robust) subset selection from the source domain. The goal is to optimize the transfer learning/broad transfer process by judiciously utilizing data from related but distinct source domains. This involves identifying and leveraging subsets of source domain data that are most beneficial for specific downstream tasks. The core challenge in both tasks lies in establishing robust methodologies to determine the relevance and adaptability of source data.

**Active data acquisition from source domain** By actively acquiring data from loosely related source domains, we could leverage the lower cost of annotation in these domains to bolster the performance on more complex downstream tasks. The strategic selection of subsets from the source domain, which offer gradients aligned with the downstream task, could optimize the transfer learning process. This approach hinges on developing robust methods for identifying which source domain data will produce gradient alignment, thereby facilitating more efficient and effective learning. Further exploration into this area may also involve understanding the limits of domain adaptability and the extent to which data from

the source domain can be used. Such advancements could significantly reduce the need for expensive labeling efforts in specialized domains and enhance the practicality of machine learning models in various applications.

**(Robust) subset selection from source domain** Understanding and quantifying the relationship between source and target domains in transfer learning and few-shot learning can be challenging, especially when they are not closely related. The research should aim to tackle this by considering various degrees of relatedness and types of data representation. More research is needed to establish criteria and develop algorithms to assess the “transferability” of source data based on how well it aligns with the gradient directions beneficial for the target task. This may include creating metrics for gradient alignment which quantify the relevance of source domain data to the target task’s learning process.

### *Integration with Vision-Language Models and Prompt Learning*

Classifying WSIs presents significant challenges due to the vast number of unlabeled patches within each slide, compounded by the availability of only slide-level labels. This scarcity of detailed labels poses substantial hurdles for both the performance and interpretability of models in histopathology. To address these issues, advanced foundation models can be leveraged with the goal of enhancing both the performance and interpretability of models in analyzing histopathology images. This direction aims to provide a more nuanced understanding and precise classification of WSIs, bridging the gap between abundant data and limited labeling.

**Improve model performance with prompt learning** Future research may explore the potential of integrating prompt learning with vision-language models to enhance whole slide image classification, particularly in few-shot learning scenarios. The proposed direction involves developing a prompt-guided pooling mechanism that leverages the Transformer ar-

chitecture’s ability to capture complex dependencies. This method could potentially allow for the prioritization and effective integration of patch-level information, aiming to extract more robust slide-level features. Such an approach might address the challenges inherent in the vast and complex nature of pathological slides, where each patch’s relevance can vary dramatically and important diagnostic features may be sparsely distributed. If successful, this strategy could not only improve the model’s discriminatory power but also contribute to more nuanced and interpretable AI-driven diagnostics. Furthermore, the adaptability of prompt learning might enable customizable tuning of the model to specific types of pathology, potentially obviating the need for extensive retraining or new data collection, thus positioning it as a candidate for rapid, efficient, and scalable deployment in clinical settings.

**Enhance visualization with prompt learning** Foundation models hold promise for enhancing the interpretability and visualization of histopathology image models. A potential research direction involves the integration of additive multiple instance learning [Javed et al., 2022] with foundation models. Additive multiple-instance learning provides a framework that not only boosts model performance but also enhances interpretability. By attributing explicit spatial credit, this approach enables a more detailed understanding of model decisions, which closely aligns with the diagnostic regions identified by pathologists and offers clearer and more relevant insights than traditional attention mechanisms. The envisioned research path could include merging additive multiple-instance learning with prompt learning to further enhance the model’s visualization and interpretative capabilities. This combined approach is anticipated to leverage the strengths of both methodologies, potentially providing deep insights into machine learning-driven diagnostics, especially useful in few-shot learning scenarios where data scarcity poses significant challenges.



## *Integration with Next-generation Molecular Profiling Technology*

The intricate spatial geometry of tissue biopsies, which is indicative of complex cellular interactions, represents a rich dataset for advancing representation learning in computational biology. The emergence of spatial transcriptomics (ST) [Ståhl et al., 2016] techniques has significantly enhanced our ability to measure RNA expressions within these cellular microenvironments, offering a novel perspective for enriching the data fidelity of single-cell RNA sequencing (scRNA-seq) and histopathology image analysis. However, current images produced by ST are often of low magnification, and the RNA expression data they yield can be marred by high levels of noise. This poses a significant hurdle in accurately interpreting the complex interplay of cellular activities and understanding the nuanced spatial relationships within tissues. Overcoming these limitations requires developing computational methods capable of extracting meaningful insights from noisy expression data and low-magnification images, thereby unlocking the full potential of ST for revolutionizing our understanding of cellular mechanisms and disease pathology.

**Improve identification of spatially varying genes and cell types** The potential refinement of integrating ST data with H&E stained images could be explored through automated region alignment and simulation of gene expression across these aligned regions within a unified algorithmic framework. This approach aims to automate the intricate process of identifying and characterizing inflammatory conditions, such as those seen in inflammatory bowel disease (IBD), across the intestinal walls. Such integration could potentially enhance the precision and efficiency of disease characterization by augmenting pathologists' expertise with machine learning capabilities. This could contribute significantly to the development of targeted therapies. Moreover, this strategy may represent a step toward automated, scalable analysis of histopathological data, potentially facilitating rapid and informed clinical decision-making.

**Improve representation learning with ST** Future research might capitalize on the dual perspectives provided by ST microscope imaging and single-cell RNA sequencing (scRNA-seq) data. This could be achieved by employing contrastive learning or multi-view learning paradigms to refine representation learning models. Such approaches have the potential to create more nuanced and informative representations that more accurately capture the biological complexity of tissue samples. The prospect of these enhanced representations revolutionizing the understanding of cellular mechanisms is considerable. Additionally, quantitatively evaluating the improvements in representations for scRNA-seq and histopathology images could establish measurable benchmarks for progress in this field. This direction could potentially pave the way for novel diagnostic and therapeutic strategies that are informed by deeper, data-driven insights.

## 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.
- Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. Support vector machines for multiple-instance learning. *Advances in neural information processing systems*, 15, 2002.
- Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra Perez, and Jorge Luis Reyes Ortiz. A public domain dataset for human activity recognition using smartphones. In *Proceedings of the 21th International European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pages 437–442, 2013.
- Antreas Antoniou and Amos Storkey. Assume, augment and learn: Unsupervised few-shot meta-learning via random labels and data augmentation. *arXiv preprint arXiv:1902.09884*, 2019.
- Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. *arXiv preprint arXiv:1906.03671*, 2019.
- Morteza Babaie, Shivam Kalra, Aditya Sriram, Christopher Mitcheltree, Shujin Zhu, Amin Khatami, Shahryar Rahnamayan, and Hamid R Tizhoosh. Classification and retrieval of digital pathology scans: A new dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 8–16, 2017.
- Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *Advances in neural information processing systems*, 32, 2019.
- Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.
- Babak Ehteshami Bejnordi, Mitko Veta, Paul Johannes Van Diest, Bram Van Ginneken, Nico Karssemeijer, Geert Litjens, Jeroen AWM Van Der Laak, Meyke Hermsen, Quirine F Manson, Maschenka Balkenhol, et al. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *Jama*, 318(22): 2199–2210, 2017.
- Sean Bell and Kavita Bala. Learning visual similarity for product design with convolutional neural networks. *ACM transactions on graphics (TOG)*, 34(4):1–10, 2015.

- Gowtham Bellala, Suresh K Bhavnani, and Clayton Scott. Extensions of generalized binary search to group identification and exponential costs. In *NIPS*, pages 154–162, 2010.
- Kaustav Bera, Kurt A Schalper, David L Rimm, Vamsidhar Velcheti, and Anant Madabhushi. Artificial intelligence in digital pathology—new tools for diagnosis and precision oncology. *Nature reviews Clinical oncology*, 16(11):703–715, 2019.
- Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks, 2015.
- 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.
- Herb Brody. Medical imaging. *Nature*, 502(7473):S81–S81, 2013.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- Carlo Brugnara, Terry Fenton, and James W Winkelman. Management training for pathology residents: I. results of a national survey. *American journal of clinical pathology*, 101(5):559–563, 1994.
- Gabriele Campanella, Matthew G Hanna, Luke Geneslaw, Allen Mirafior, 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)*, pages 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.
- Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.
- Venkatesan T Chakaravarthy, Vinayaka Pandit, Sambuddha Roy, Pranjal Awasthi, and Mukesh Mohania. Decision trees for entity identification: Approximation algorithms and hardness results. In *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 53–62. ACM, 2007.

- Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. *Advances in neural information processing systems*, 32, 2019.
- 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*, pages 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*, pages 4015–4025, 2021b.
- Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In *International conference on machine learning*, pages 1683–1691. PMLR, 2014.
- Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020a.
- Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020b.
- 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.
- Xiaocong Chen, Lina Yao, Tao Zhou, Jinming Dong, and Yu Zhang. Momentum contrastive learning for few-shot covid-19 diagnosis from chest ct images. *Pattern recognition*, 113: 107826, 2021c.
- Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021.
- Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015a.
- Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020c.
- Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9640–9649, 2021d.

- Yuxin Chen and Andreas Krause. Near-optimal batch mode active learning and adaptive submodular optimization. In *International Conference on Machine Learning (ICML)*, June 2013a.
- Yuxin Chen and Andreas Krause. Near-optimal batch mode active learning and adaptive submodular optimization. In *International Conference on Machine Learning*, pages 160–168. PMLR, 2013b.
- Yuxin Chen, S. Hamed Hassani, Amin Karbasi, and Andreas Krause. Sequential information maximization: When is greedy near-optimal? In *Proc. International Conference on Learning Theory (COLT)*, July 2015b.
- Yuxin Chen, S Hamed Hassani, Amin Karbasi, and Andreas Krause. Sequential information maximization: When is greedy near-optimal? In *Conference on Learning Theory*, pages 338–363. PMLR, 2015c.
- Yuxin Chen, S. Hamed Hassani, and Andreas Krause. Near-optimal bayesian active learning with correlated and noisy tests, 2016.
- Yuxin Chen, Jean-Michel Renders, Morteza Haghiri Chehreghani, and Andreas Krause. Efficient online learning for optimizing value of information: Theory and application to interactive troubleshooting. *arXiv preprint arXiv:1703.05452*, 2017.
- Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005.
- Gui Citovsky, Giulia DeSalvo, Claudio Gentile, Lazaros Karydas, Anand Rajagopalan, Afshin Rostamizadeh, and Sanjiv Kumar. Batch active learning at scale. *Advances in Neural Information Processing Systems*, 34:11933–11944, 2021.
- Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. Emnist: an extension of mnist to handwritten letters. *arXiv preprint arXiv:1702.05373*, 2017.
- David A Cohn, Zoubin Ghahramani, and Michael I Jordan. Active learning with statistical models. *Journal of artificial intelligence research*, 4:129–145, 1996.
- Nicolas Coudray, Paolo Santiago Ocampo, Theodore Sakellaropoulos, Navneet Narula, Matija Snuderl, David Fenyö, Andre L Moreira, Narges Razavian, and Aristotelis Tsirigos. Classification and mutation prediction from non-small cell lung cancer histopathology images using deep learning. *Nature medicine*, 24(10):1559–1567, 2018.
- Luke N Darlow, Elliot J Crowley, Antreas Antoniou, and Amos J Storkey. Cinc-10 is not imagenet or cifar-10. *arXiv preprint arXiv:1810.03505*, 2018.
- Sanjoy Dasgupta. Analysis of a greedy active learning strategy. *Advances in neural information processing systems*, 17:337–344, 2005.

- Sanjoy Dasgupta and J Langford. Active learning. *Encyclopedia of Machine Learning*, 2011.
- 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.
- Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*, pages 376–380, 2014.
- Akshay Raj Dhamija, Touqeer Ahmad, Jonathan Schwan, Mohsen Jafarzadeh, Chunchun Li, and Terrance E Boulton. Self-supervised features improve open-world learning. *arXiv preprint arXiv:2102.07848*, 2021.
- Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1-2):31–71, 1997.
- Nan Ding, Youhan Fang, Ryan Babbush, Changyou Chen, Robert D Skeel, and Hartmut Neven. Bayesian sampling using stochastic gradient thermostats. *Advances in neural information processing systems*, 27, 2014.
- 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*, pages 647–655. PMLR, 2014.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- 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.
- Melanie Ducoffe and Frederic Precioso. Adversarial active learning for deep networks: a margin based approach. *arXiv preprint arXiv:1802.09841*, 2018.
- 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*, pages 1126–1135. PMLR, 2017.
- Andrew Foong, David Burt, Yingzhen Li, and Richard Turner. On the expressiveness of approximate inference in bayesian neural networks. *Advances in Neural Information Processing Systems*, 33:15897–15908, 2020.

- Vincent Fortuin, Adrià Garriga-Alonso, Florian Wenzel, Gunnar Rätsch, Richard Turner, Mark van der Wilk, and Laurence Aitchison. Bayesian neural network priors revisited. *arXiv preprint arXiv:2102.06571*, 2021.
- Linton C Freeman and Linton C Freeman. *Elementary applied statistics: for students in behavioral science*. New York: Wiley, 1965.
- Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.
- Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *International conference on machine learning*, pages 1183–1192. PMLR, 2017.
- Jhair Gallardo, Tyler L Hayes, and Christopher Kanan. Self-supervised training enhances online continual learning. *arXiv preprint arXiv:2103.14010*, 2021.
- Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. *Advances in neural information processing systems*, 31, 2018a.
- Octavian Ganea, Gary Becigneul, and Thomas Hofmann. Hyperbolic neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 5345–5355. Curran Associates, Inc., 2018b. URL <http://papers.nips.cc/paper/7780-hyperbolic-neural-networks.pdf>.
- Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.
- Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. Boosting few-shot visual learning with self-supervision. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8059–8068, 2019.
- Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.
- Daniel Golovin, Andreas Krause, and Debajyoti Ray. Near-optimal bayesian active learning with noisy observations. *arXiv preprint arXiv:1010.3091*, 2010.
- Teofilo F Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical computer science*, 38:293–306, 1985.
- 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.



- Robert L Grossman, Allison P Heath, Vincent Ferretti, Harold E Varmus, Douglas R Lowy, Warren A Kibbe, and Louis M Staudt. Toward a shared vision for cancer genomic data. *New England Journal of Medicine*, 375(12):1109–1112, 2016.
- Carlos Guestrin, Andreas Krause, and Ajit Paul Singh. Near-optimal sensor placements in gaussian processes. In *Proceedings of the 22nd international conference on Machine learning*, pages 265–272, 2005.
- Caglar Gulcehre, Misha Denil, Mateusz Malinowski, Ali Razavi, Razvan Pascanu, Karl Moritz Hermann, Peter Battaglia, Victor Bapst, David Raposo, Adam Santoro, et al. Hyperbolic attention networks. *arXiv preprint arXiv:1805.09786*, 2018.
- Guy Hacoheh, Avihu Dekel, and Daphna Weinshall. Active learning on a budget: Opposite strategies suit high and low budgets. *arXiv preprint arXiv:2202.02794*, 2022.
- Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.
- 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*, pages 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*, pages 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*, pages 9729–9738, 2020.
- Xiangteng He and Yuxin Peng. Fine-grained visual-textual representation learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(2):520–531, 2019.
- 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*, pages 4182–4192. PMLR, 2020.
- José Miguel Hernández-Lobato and Ryan Adams. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *International conference on machine learning*, pages 1861–1869. PMLR, 2015.

- R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.
- Chih-Hui Ho, Pedro Morgado, Amir Persekian, and Nuno Vasconcelos. Pies: Pose invariant embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12377–12386, 2019.
- Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- Steven CH Hoi, Rong Jin, and Michael R Lyu. Large-scale text categorization by batch mode active learning. In *Proceedings of the 15th international conference on World Wide Web*, pages 633–642, 2006a.
- Steven CH Hoi, Rong Jin, Jianke Zhu, and Michael R Lyu. Batch mode active learning and its application to medical image classification. In *Proceedings of the 23rd international conference on Machine learning*, pages 417–424, 2006b.
- Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.
- Kyle Hsu, Sergey Levine, and Chelsea Finn. Unsupervised learning via meta-learning. *arXiv preprint arXiv:1810.02334*, 2018.
- Wei-Ning Hsu and Hsuan-Tien Lin. Active learning by learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2015.
- Jiaji Huang, Rewon Child, Vinay Rao, Hairong Liu, Sanjeev Satheesh, and Adam Coates. Active learning for speech recognition: the power of gradients. *arXiv preprint arXiv:1612.03226*, 2016.
- Sheng-Jun Huang, Rong Jin, and Zhi-Hua Zhou. Active learning by querying informative and representative examples. *Advances in neural information processing systems*, 23, 2010.
- Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, and Nicholas Carlini. Rohan 377 taori, achal dave, vaishaal shankar, hongseok namkoong, john miller, hannaneh hajishirzi, 378 ali farhadi, and ludwig schmidt. *Openclip, July*, 1(2):4, 2021.
- Gabriel Ilharco, Mitchell Wortsman, Samir Yitzhak Gadre, Shuran Song, Hannaneh Hajishirzi, Simon Kornblith, Ali Farhadi, and Ludwig Schmidt. Patching open-vocabulary models by interpolating weights. *Advances in Neural Information Processing Systems*, 35: 29262–29277, 2022.
- Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *International conference on machine learning*, pages 2127–2136. PMLR, 2018.

- Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.
- Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silvana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, Jayne Seekins, David A. Mong, Safwan S. Halabi, Jesse K. Sandberg, Ricky Jones, David B. Larson, Curtis P. Langlotz, Bhavik N. Patel, Matthew P. Lungren, and Andrew Y. Ng. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison, 2019.
- Pavel Izmailov, Sharad Vikram, Matthew D Hoffman, and Andrew Gordon Gordon Wilson. What are bayesian neural network posteriors really like? In *International conference on machine learning*, pages 4629–4640. PMLR, 2021.
- David Janz, Jos van der Westhuizen, and José Miguel Hernández-Lobato. Actively learning what makes a discrete sequence valid. *arXiv preprint arXiv:1708.04465*, 2017.
- Shervin Javdani, Yuxin Chen, Amin Karbasi, Andreas Krause, Drew Bagnell, and Siddhartha Srinivasa. Near optimal bayesian active learning for decision making. In *Artificial Intelligence and Statistics*, pages 430–438. PMLR, 2014.
- 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.
- Mark A Jensen, Vincent Ferretti, Robert L Grossman, and Louis M Staudt. The nci genomic data commons as an engine for precision medicine. *Blood, The Journal of the American Society of Hematology*, 130(4):453–459, 2017.
- Baoyu Jing, Pengtao Xie, and Eric Xing. On the automatic generation of medical imaging reports. *arXiv preprint arXiv:1711.08195*, 2017.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.
- Matti Kääriäinen. Active learning in the non-realizable case. In *International Conference on Algorithmic Learning Theory*, pages 63–77. Springer, 2006.
- Athresh Karanam, Krishnateja Killamsetty, Harsha Kokel, and Rishabh K Iyer. Orient: Submodular mutual information measures for data subset selection under distribution shift. *Advances in neural information processing systems*, 2022.
- Jakob Nikolas Kather, Cleo-Aron Weis, Francesco Bianconi, Susanne M Melchers, Lothar R Schad, Timo Gaiser, Alexander Marx, and Frank Gerrit Zöllner. Multi-class texture analysis in colorectal cancer histology. *Scientific reports*, 6(1):1–11, 2016.

- 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>.
- Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv preprint arXiv:1511.02680*, 2015.
- Ashish Kheta, Zachary C Lipton, and Anima Anandkumar. Learning from noisy singly-labeled data. *arXiv preprint arXiv:1712.04577*, 2017.
- Siavash Khodadadeh, Ladislau Boloni, and Mubarak Shah. Unsupervised meta-learning for few-shot image classification. *Advances in neural information processing systems*, 32, 2019.
- Siavash Khodadadeh, Sharare Zehtabian, Saeed Vahidian, Weijia Wang, Bill Lin, and Ladislau Bölöni. Unsupervised meta-learning through latent-space interpolation in generative models. *arXiv preprint arXiv:2006.10236*, 2020.
- Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschiot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.
- Valentin Khrulkov, Leyla Mirvakhabova, Evgeniya Ustinova, Ivan V. Oseledets, and Victor S. Lempitsky. Hyperbolic image embeddings. *CoRR*, abs/1904.02239, 2019. URL <http://arxiv.org/abs/1904.02239>.
- Valentin Khrulkov, Leyla Mirvakhabova, Evgeniya Ustinova, Ivan Oseledets, and Victor Lempitsky. Hyperbolic image embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6418–6428, 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.
- Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.
- Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- Diederik P Kingma, Danilo J Rezende, Shakir Mohamed, and Max Welling. Semi-supervised learning with deep generative models. *arXiv preprint arXiv:1406.5298*, 2014.
- Diederik P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. *arXiv preprint arXiv:1506.02557*, 2015.

- Andreas Kirsch, Joost Van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. *Advances in neural information processing systems*, 32, 2019.
- Andreas Kirsch, Sebastian Farquhar, and Yarin Gal. A simple baseline for batch active learning with stochastic acquisition functions. *arXiv preprint arXiv:2106.12059*, 2021a.
- Andreas Kirsch, Tom Rainforth, and Yarin Gal. Test distribution-aware active learning: A principled approach against distribution shift and outliers. *arXiv preprint arXiv:2106.11719*, 2021b.
- Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille, 2015.
- Murat Koklu and Ilker Ali Ozkan. Multiclass classification of dry beans using computer vision and machine learning techniques. *Computers and Electronics in Agriculture*, 174: 105507, 2020.
- Suraj Kothawade, Nathan Beck, Krishnateja Killamsetty, and Rishabh Iyer. Similar: Sub-modular information measures based active learning in realistic scenarios. *Advances in Neural Information Processing Systems*, 34:18685–18697, 2021.
- Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Citeseer*, 2009.
- Girish Kulkarni, Visruth Premraj, Vicente Ordonez, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. Babytalk: Understanding and generating simple image descriptions. *IEEE transactions on pattern analysis and machine intelligence*, 35 (12):2891–2903, 2013.
- Yann Le Cun and Françoise Fogelman-Soulié. Modèles connexionnistes de l’apprentissage. *Intellectica*, 2(1):114–143, 1987.
- Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- Dong Bok Lee, Dongchan Min, Seanie Lee, and Sung Ju Hwang. Meta-gmvae: Mixture of gaussian vae for unsupervised meta-learning. In *International Conference on Learning Representations*, 2020.
- Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiosek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International Conference on Machine Learning*, pages 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*, pages 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*, pages 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*, pages 14318–14328, 2021.
- Chunyuan Li, Changyou Chen, David Carlson, and Lawrence Carin. Preconditioned stochastic gradient langevin dynamics for deep neural networks. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016a.
- Chunyuan Li, Andrew Stevens, Changyou Chen, Yunchen Pu, Zhe Gan, and Lawrence Carin. Learning weight uncertainty with stochastic gradient mcmc for shape classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5666–5675, 2016b.
- 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*, pages 1–10, 2019.
- Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.
- 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.
- Dennis V Lindley. On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, 27(4):986–1005, 1956.
- Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.
- Geert Litjens, Peter Bandi, Babak Ehteshami Bejnordi, Oscar Geessink, Maschenka Balkenhol, Peter Bult, Altuna Halilovic, Meyke Hermsen, Rob Van de Loo, Rob Vogels, et al. 1399 h&e-stained sentinel lymph node sections of breast cancer patients: the camelyon dataset. *GigaScience*, 7(6):giy065, 2018.
- Chenxi Liu, Junhua Mao, Fei Sha, and Alan Yuille. Attention correctness in neural image captioning. In *Proceedings of the AAAI conference on artificial intelligence*, 2017a.
- Qiang Liu, Zhaocheng Liu, Xiaofang Zhu, and Yeliang Xiu. Deep active learning by model interpretability. *arXiv preprint arXiv:2007.12100*, 2020.

- Yun Liu, Krishna Gadepalli, Mohammad Norouzi, George E Dahl, Timo Kohlberger, Aleksey Boyko, Subhashini Venugopalan, Aleksei Timofeev, Philip Q Nelson, Greg S Corrado, et al. Detecting cancer metastases on gigapixel pathology images. *arXiv preprint arXiv:1703.02442*, 2017b.
- 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.
- Yuning Lu, Liangjian Wen, Jianzhuang Liu, Yajing Liu, and Xinmei Tian. Self-supervision can be a good few-shot learner. In *European conference on computer vision*, pages 740–758. Springer, 2022.
- David JC MacKay. Information-based objective functions for active data selection. *Neural computation*, 4(4):590–604, 1992.
- Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. *Advances in Neural Information Processing Systems*, 32, 2019.
- Kushagra Mahajan, Monika Sharma, and Lovekesh Vig. Meta-dermdiagnosis: Few-shot skin disease identification using meta-learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 730–731, 2020.
- Puneet Mangla, Nupur Kumari, Abhishek Sinha, Mayank Singh, Balaji Krishnamurthy, and Vineeth N Balasubramanian. Charting the right manifold: Manifold mixup for few-shot learning. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2218–2227, 2020.
- Alfonso Medela, Artzai Picon, Cristina L Saratxaga, Oihana Belar, Virginia Cabezón, Riccardo Cicchi, Roberto Bilbao, and Ben Glover. Few shot learning in histopathological images: reducing the need of labeled data on biological datasets. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pages 1860–1864. IEEE, 2019.
- Carlos Medina, Arnout Devos, and Matthias Grossglauser. Self-supervised prototypical transfer learning for few-shot classification. *arXiv preprint arXiv:2006.11325*, 2020.

- 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*, pages 6707–6717, 2020.
- Sudhanshu Mittal, Maxim Tatarchenko, Özgün Çiçek, and Thomas Brox. Parting with illusions about deep active learning. *arXiv preprint arXiv:1912.05361*, 2019.
- Pooya Mobadersany, Safoora Yousefi, Mohamed Amgad, David A Gutman, Jill S Barnholtz-Sloan, José E Velázquez Vega, Daniel J Brat, and Lee AD Cooper. Predicting cancer outcomes from histology and genomics using convolutional networks. *Proceedings of the National Academy of Sciences*, 115(13):E2970–E2979, 2018.
- Yair Movshovitz-Attias, Alexander Toshev, Thomas K Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *Proceedings of the IEEE international conference on computer vision*, pages 360–368, 2017.
- Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *International conference on machine learning*, pages 2554–2563. PMLR, 2017.
- 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*, pages 329–338. Springer, 2021.
- Mohammad Naghshvar, Tara Javidi, and Kamalika Chaudhuri. Noisy bayesian active learning. In *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1626–1633. IEEE, 2012.
- Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.
- George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1): 265–294, 1978.
- Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.
- Alex Nichol and John Schulman. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2(3):4, 2018.
- Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 30, 2017.
- Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4004–4012, 2016.



- Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.
- Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.
- 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.
- Natalia Ostapuk, Jie Yang, and Philippe Cudré-Mauroux. Activelink: deep active learning for link prediction in knowledge graphs. In *The World Wide Web Conference*, pages 1398–1408, 2019.
- 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.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.
- 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*, pages 9052–9061, 2021.
- Tiexin Qin, Wenbin Li, Yinghuan Shi, and Yang Gao. Diversity helps: Unsupervised few-shot learning via distribution shift-based data augmentation. *arXiv preprint arXiv:2004.05805*, 2020.
- Joaquin Quinonero-Candela, Carl Edward Rasmussen, Fabian Sinz, Olivier Bousquet, and Bernhard Schölkopf. Evaluating predictive uncertainty challenge. In *Machine Learning Challenges Workshop*, pages 1–27. Springer, 2005.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 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.

- Antti Rasmus, Harri Valpola, Mikko Honkala, Mathias Berglund, and Tapani Raiko. Semi-supervised learning with ladder networks. *arXiv preprint arXiv:1507.02672*, 2015.
- Alexander Ratner, Braden Hancock, Jared Dunnmon, Frederic Sala, Shreyash Pandey, and Christopher Ré. Training complex models with multi-task weak supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4763–4771, 2019.
- Alexander J Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. Data programming: Creating large training sets, quickly. *Advances in neural information processing systems*, 29, 2016.
- Pierre H Richemond, Jean-Bastien Grill, Florent Alché, 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.
- Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th international conference on international conference on machine learning*, pages 833–840, 2011.
- Hippolyt Ritter, Aleksandar Botev, and David Barber. A scalable laplace approximation for neural networks. In *6th International Conference on Learning Representations, ICLR 2018-Conference Track Proceedings*, volume 6. International Conference on Representation Learning, 2018.
- Joshua Robinson, Stefanie Jegelka, and Suvrit Sra. Strength from weakness: Fast learning using weak supervision. In *International Conference on Machine Learning*, pages 8127–8136. PMLR, 2020.
- Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- Nicholas Roy and Andrew McCallum. Toward optimal active learning through monte carlo estimation of error reduction. *ICML, Williamstown*, 2:441–448, 2001.
- Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015a. doi:10.1007/s11263-015-0816-y.
- Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015b.
- 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, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850. PMLR, 2016.
- 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.
- Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.
- Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017.
- Burr Settles. Active learning: Synthesis lectures on artificial intelligence and machine learning. *Long Island, NY: Morgan & Clay Pool*, 10:S00429ED1V01Y201207AIM018, 2012.
- 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.
- Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.
- 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*, pages 682–698. PMLR, 2021.
- Yanyao Shen, Hyokun Yun, Zachary C Lipton, Yakov Kronrod, and Animashree Anandkumar. Deep active learning for named entity recognition. *arXiv preprint arXiv:1707.05928*, 2017.
- 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.
- Milad Sikaroudi, Amir Safarpour, Benyamin Ghogh, Sobhan Shafiei, Mark Crowley, and Hamid R Tizhoosh. Supervision and source domain impact on representation learning: A histopathology case study. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 1400–1403. IEEE, 2020.

- Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5972–5981, 2019.
- Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.
- Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29, 2016.
- 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. Mastrogiannakis 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.
- Patrik L Ståhl, Fredrik Salmén, Sanja Vickovic, Anna Lundmark, José Fernández Navarro, Jens Magnusson, Stefania Giacomello, Michaela Asp, Jakub O Westholm, Mikael Huss, et al. Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. *Science*, 353(6294):78–82, 2016.
- 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*, pages 945–953, 2015.
- Jong-Chyi Su, Subhransu Maji, and Bharath Hariharan. When does self-supervision improve few-shot learning? In *European conference on computer vision*, pages 645–666. Springer, 2020.
- Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017.
- 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*, pages 1199–1208, 2018.
- Wei Tan, Lan Du, and Wray Buntine. Diversity enhanced active learning with strictly proper scoring rules. *Advances in Neural Information Processing Systems*, 34:10906–10918, 2021.

- Eu Wern Teh and Graham W Taylor. Learning with less data via weakly labeled patch classification in digital pathology. In *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, pages 471–475. IEEE, 2020.
- Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 776–794. Springer, 2020a.
- Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Re-thinking 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*, pages 266–282. Springer, 2020b.
- Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001.
- Eleni Triantafillou, Richard Zemel, and Raquel Urtasun. Few-shot learning through an information retrieval lens. *Advances in neural information processing systems*, 30, 2017.
- Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, et al. Meta-dataset: A dataset of datasets for learning to learn from few examples. *arXiv preprint arXiv:1903.03096*, 2019.
- Tobias Uelwer, Jan Robine, Stefan Sylvius Wagner, Marc Höftmann, Eric Upschulte, Sebastian Konietzny, Maike Behrendt, and Stefan Harmeling. A survey on self-supervised representation learning. *arXiv preprint arXiv:2308.11455*, 2023.
- Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.
- 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.
- Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015.
- Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant cnns for digital pathology. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16–20, 2018, Proceedings, Part II 11*, pages 210–218. Springer, 2018.

- Alexander Vergara, Shankar Vembu, Tuba Ayhan, Margaret A Ryan, Margie L Homer, and Ramón Huerta. Chemical gas sensor drift compensation using classifier ensembles. *Sensors and Actuators B: Chemical*, 166:320–329, 2012.
- Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010.
- Sagar Vinodababu. a-pytorch-tutorial-to-image-captioning. <https://github.com/sgrvino/d/a-PyTorch-Tutorial-to-Image-Captioning>, 2019.
- Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. Order matters: Sequence to sequence for sets. *arXiv preprint arXiv:1511.06391*, 2015a.
- Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015b.
- Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.
- Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1386–1393, 2014.
- Keze Wang, Dongyu Zhang, Ya Li, Ruimao Zhang, and Liang Lin. Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(12):2591–2600, 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*, pages 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*, pages 7278–7286, 2018.
- Zheng Wang and Jieping Ye. Querying discriminative and representative samples for batch mode active learning. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 9(3):1–23, 2015.
- Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of machine learning research*, 10(2), 2009.

- Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688. Citeseer, 2011.
- Florian Wenzel, Kevin Roth, Bastiaan S Veeling, Jakub Świątkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. How good is the bayes posterior in deep neural networks really? *arXiv preprint arXiv:2002.02405*, 2020.
- Zhirong Wu, Alexei A Efros, and Stella X Yu. Improving generalization via scalable neighborhood component analysis. In *Proceedings of the european conference on computer vision (ECCV)*, pages 685–701, 2018a.
- 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*, pages 3733–3742, 2018b.
- 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.
- Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.
- Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015.
- 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.
- Yi-Fan Yan, Sheng-Jun Huang, Shaoyi Chen, Meng Liao, and Jin Xu. Active learning with query generation for cost-effective text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6583–6590, 2020.
- 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.
- Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8808–8817, 2020.
- Byung-Jun Yoon, Xiaoning Qian, and Edward R Dougherty. Quantifying the objective cost of uncertainty in complex dynamical systems. *IEEE Transactions on Signal Processing*, 61(9):2256–2266, 2013.
- Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4651–4659, 2016.
- 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.
- Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International conference on machine learning*, pages 12310–12320. PMLR, 2021.
- 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*, pages 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)*, pages 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*, pages 418–435. PMLR, 2020b.
- Renyu Zhang, Aly A Khan, Yuxin Chen, and Robert L Grossman. Enhancing instance-level image classification with set-level labels. *arXiv preprint arXiv:2311.05659*, 2023a.
- Renyu Zhang, Aly A Khan, Robert L Grossman, and Yuxin Chen. Scalable batch-mode deep bayesian active learning via equivalence class annealing. In *The Eleventh International Conference on Learning Representations*, 2023b.



- Ruqi Zhang, Chunyuan Li, Jianyi Zhang, Changyou Chen, and Andrew Gordon Wilson. Cyclical stochastic gradient mcmc for bayesian deep learning. *arXiv preprint arXiv:1902.03932*, 2019.
- Zizhao Zhang, Yuanpu Xie, Fuyong Xing, Mason McGough, and Lin Yang. Mdnet: A semantically and visually interpretable medical image diagnosis network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6428–6436, 2017.
- Guang Zhao, Edward Dougherty, Byung-Jun Yoon, Francis Alexander, and Xiaoning Qian. Uncertainty-aware active learning for optimal bayesian classifier. In *International Conference on Learning Representations (ICLR 2021)*, 2021.
- Alice X Zheng, Irina Rish, and Alina Beygelzimer. Efficient test selection in active diagnosis via entropy approximation. *arXiv preprint arXiv:1207.1418*, 2012.
- Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021.
- Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National science review*, 5(1):44–53, 2018.
- Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6002–6012, 2019.

## APPENDIX A

### APPENDIX FOR ENHANCING INSTANCE-LEVEL IMAGE CLASSIFICATION WITH SET-LEVEL LABELS

#### A.1 Training Details

##### *A.1.1 Pretrain with Unique Class Number and Most Frequent Class of Input Sets*

In our study, an epoch refers to going through all the input sets in the dataset once. 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 and a temperature of 0.07. 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.

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.

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.  $\ell_1$  loss is optimized for pretraining with unique class numbers of input sets. For FACILE-SupCon, we train the model with 2,000 epochs and a batch size of 256. An additional temperature parameter is set to 0.07. The SGD is used with a learning rate of 0.05, weight decay of  $1e-4$ , and momentum of 0.9.

##### *A.1.2 Fine-tune ViT-B/16 of CLIP with CUB200*

SimSiam is trained for 400 epochs using a batch size of 64. SGD is used with an initial learning rate of 0.0125, weight decay of  $1e-4$ , and momentum of 0.9. The cosine scheduler is

used for the optimizer. SimCLR is also trained for 400 epochs with a batch size of 64. An additional temperature parameter is set to 0.07. SGD is used with a learning rate of 0.0125, weight decay of  $1e-4$ , and momentum of 0.9. The training also uses a cosine scheduler.

FACILE-FSP is trained for 200 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. For FACILE-SupCon, we train the model with 800 epochs and a batch size of 64. An additional temperature parameter is set to 0.07. The SGD is used with an initial learning rate of 0.0125, weight decay of  $1e-4$ , and momentum of 0.9. Both models’ training utilized a cosine annealing scheduler.

### *A.1.3 Pretrain ResNet18 with TCGA and GTEx Dataset*

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. An additional temperature parameter is set to 0.07. 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 3,000 epochs with batch size 32. 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 training process is configured with a batch size of 32 and extends over 3,000 epochs. An additional temperature parameter is set to 0.07. We use SGD with a learning rate of 0.00625, 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.

#### *A.1.4 Fine-tune ViT-B/14 of DINO V2 with TCGA*

SimSiam is trained for 400 epochs with a batch size of 64, utilizing Stochastic Gradient Descent (SGD) with an initial learning rate of 0.0125, a weight decay of 1e-4, and a momentum of 0.9. A cosine scheduler was employed. SimCLR underwent a similar training regimen for 400 epochs and a batch size of 64, with an additional temperature parameter set at 0.07 and identical SGD parameters, including the use of a cosine scheduler for learning rate adjustments.

FSP-Patch also completed 400 epochs of training with a batch size of 64. The model employed SGD with a learning rate of 0.0125, a weight decay of 1e-4, and a momentum of 0.9, along with a cosine scheduler to modulate the learning rate.

For FACILE-FSP, training spanned 200 epochs with a batch size of 64, using SGD with the same learning rate, weight decay, and momentum settings. FACILE-SupCon extended its training to 800 epochs with a batch size of 64, including an additional temperature setting of 0.07 and the same SGD configuration. Both FACILE-FSP and FACILE-SupCon models utilized a cosine annealing scheduler.

## A.2 Additional Result

### A.2.1 Pretrain ResNet18 on TCGA with Patch Size 224X224

ACC on LC, PAIP, and NCT Datasets

We pretrain the models on TCGA datasets with patches size  $224 \times 224$  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 A.1.

Test with Large Shot Number

We further test the trained models with a larger shot number  $k$ . The result is shown in Table A.2

### A.2.2 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 dataset, which are also studied in Yang et al. [2022].

The SimSiam model is trained for 100 epochs. SGD optimizer is used with a 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., 2021d, 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 a learning rate of 0.5, momentum of 0.9, and weight decay of 0 are used. A

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	76.64 ± 0.50	77.88 ± 0.47	76.77 ± 0.47	77.15 ± 0.48	<b>77.16 ± 0.48</b>
FACILE-FSP	<b>79.01 ± 0.49</b>	<b>78.16 ± 0.48</b>	<b>77.43 ± 0.50</b>	<b>79.15 ± 0.47</b>	75.81 ± 0.48
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	91.16 ± 0.24	90.48 ± 0.24	90.40 ± 0.24	91.39 ± 0.24	<b>91.03 ± 0.22</b>
FACILE-FSP	<b>91.77 ± 0.21</b>	<b>90.85 ± 0.23</b>	<b>90.77 ± 0.24</b>	<b>92.19 ± 0.22</b>	90.02 ± 0.24
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	<b>51.61 ± 0.68</b>	<b>51.61 ± 0.67</b>	<b>52.06 ± 0.67</b>	<b>51.74 ± 0.66</b>	<b>51.38 ± 0.66</b>
FACILE-SupCon	49.65 ± 0.61	51.32 ± 0.66	51.16 ± 0.63	50.00 ± 0.62	50.81 ± 0.65
FACILE-FSP	48.91 ± 0.61	49.57 ± 0.63	49.68 ± 0.63	49.42 ± 0.65	48.60 ± 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	<b>65.44 ± 0.51</b>	<b>66.75 ± 0.52</b>	<b>67.11 ± 0.51</b>	67.24 ± 0.53	<b>67.06 ± 0.52</b>
FACILE-FSP	64.68 ± 0.53	65.75 ± 0.49	66.58 ± 0.51	<b>67.42 ± 0.53</b>	<b>67.06 ± 0.53</b>
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	<b>68.99 ± 0.45</b>	<b>70.76 ± 0.40</b>	<b>70.89 ± 0.41</b>	<b>70.45 ± 0.45</b>	70.63 ± 0.44
FACILE-FSP	67.43 ± 0.44	68.45 ± 0.4	68.97 ± 0.42	69.53 ± 0.43	<b>70.89 ± 0.42</b>
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	<b>86.89 ± 0.22</b>	<b>88.06 ± 0.20</b>	<b>89.26 ± 0.19</b>	<b>89.62 ± 0.19</b>	<b>88.67 ± 0.21</b>
FACILE-FSP	84.83 ± 0.24	85.78 ± 0.23	87.68 ± 0.20	88.16 ± 0.20	87.67 ± 0.20

Table A.1: Models tested on LC, PAIP, and NCT dataset; average ACC and CI are reported.

pretraining method	NC	LR	RC	LR+LA	RC+LA
10-shot 5-way on LC					
ImageNet (FSP)	78.76 $\pm$ 0.94	78.92 $\pm$ 0.92	80.45 $\pm$ 0.87	82.25 $\pm$ 0.83	80.20 $\pm$ 0.89
SimSiam	88.52 $\pm$ 0.55	87.20 $\pm$ 0.58	87.73 $\pm$ 0.56	91.62 $\pm$ 0.46	91.88 $\pm$ 0.47
SimCLR	87.02 $\pm$ 0.64	86.26 $\pm$ 0.64	85.61 $\pm$ 0.72	90.28 $\pm$ 0.52	89.60 $\pm$ 0.58
FSP-Patch	88.41 $\pm$ 0.53	88.64 $\pm$ 0.52	89.15 $\pm$ 0.51	90.49 $\pm$ 0.50	89.88 $\pm$ 0.54
FACILE-SupCon	92.84 $\pm$ 0.39	92.87 $\pm$ 0.38	93.21 $\pm$ 0.37	94.25 $\pm$ 0.36	<b>93.72 <math>\pm</math> 0.39</b>
FACILE-FSP	<b>93.10 <math>\pm</math> 0.39</b>	<b>93.11 <math>\pm</math> 0.38</b>	<b>93.63 <math>\pm</math> 0.37</b>	<b>94.52 <math>\pm</math> 0.35</b>	93.07 $\pm$ 0.45
10-shot 3-way on PAIP					
ImageNet (FSP)	65.36 $\pm$ 0.91	65.17 $\pm$ 1.00	65.40 $\pm$ 0.99	66.52 $\pm$ 0.81	64.45 $\pm$ 0.81
SimSiam	67.19 $\pm$ 0.88	67.35 $\pm$ 0.98	68.55 $\pm$ 0.94	70.88 $\pm$ 0.77	70.62 $\pm$ 0.77
SimCLR	65.77 $\pm$ 0.85	66.70 $\pm$ 0.91	67.01 $\pm$ 0.91	68.41 $\pm$ 0.79	66.96 $\pm$ 0.82
FSP-Patch	68.50 $\pm$ 0.82	69.12 $\pm$ 0.85	69.39 $\pm$ 0.85	70.13 $\pm$ 0.75	68.25 $\pm$ 0.76
FACILE-SupCon	<b>70.03 <math>\pm</math> 0.81</b>	<b>71.24 <math>\pm</math> 0.84</b>	72.17 $\pm$ 0.83	<b>73.31 <math>\pm</math> 0.71</b>	72.50 $\pm$ 0.71
FACILE-FSP	69.19 $\pm$ 0.82	71.13 $\pm$ 0.82	71.78 $\pm$ 0.81	73.22 $\pm$ 0.73	<b>72.78 <math>\pm</math> 0.71</b>
10-shot 9-way on NCT					
ImageNet (FSP)	78.76 $\pm$ 0.94	78.92 $\pm$ 0.92	80.45 $\pm$ 0.87	82.25 $\pm$ 0.83	80.20 $\pm$ 0.89
SimSiam	82.92 $\pm$ 0.91	83.42 $\pm$ 0.89	84.76 $\pm$ 0.81	87.66 $\pm$ 0.72	88.12 $\pm$ 0.69
SimCLR	80.34 $\pm$ 0.96	81.67 $\pm$ 0.90	83.09 $\pm$ 0.84	85.96 $\pm$ 0.76	86.82 $\pm$ 0.72
FSP-Patch	83.36 $\pm$ 0.77	84.05 $\pm$ 0.74	85.93 $\pm$ 0.65	87.15 $\pm$ 0.62	86.05 $\pm$ 0.63
FACILE-SupCon	<b>89.57 <math>\pm</math> 0.49</b>	<b>91.11 <math>\pm</math> 0.45</b>	<b>92.20 <math>\pm</math> 0.41</b>	<b>92.88 <math>\pm</math> 0.39</b>	<b>92.02 <math>\pm</math> 0.41</b>
FACILE-FSP	87.54 $\pm$ 0.61	89.25 $\pm$ 0.56	90.77 $\pm$ 0.49	91.63 $\pm$ 0.48	91.23 $\pm$ 0.50

Table A.2: Test result on LC, PAIP, and NCT dataset with shot number 10; average F1 and CI are reported.

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, 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 A.3 shows the performance of the pretrained models on the LC and PAIP datasets 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 5.3.

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)	<b>68.00 ± 1.29</b>	<b>66.17 ± 1.41</b>	<b>66.18 ± 1.46</b>	<b>68.39 ± 1.34</b>	<b>68.02 ± 1.40</b>
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)	<b>83.53 ± 0.79</b>	<b>80.81 ± 1.01</b>	<b>80.27 ± 1.08</b>	<b>85.57 ± 0.77</b>	<b>84.06 ± 0.89</b>
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	<b>49.38 ± 1.19</b>
SupCon	<b>48.45 ± 1.19</b>	<b>49.29 ± 1.20</b>	<b>48.97 ± 1.22</b>	<b>49.47 ± 1.20</b>	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	<b>64.82 ± 0.86</b>	<b>64.60 ± 0.87</b>
SupCon	<b>63.62 ± 0.91</b>	<b>64.38 ± 0.96</b>	<b>63.61 ± 1.00</b>	64.37 ± 0.87	64.28 ± 0.88

Table A.3: pretraining on NCT dataset and testing on LC and PAIP dataset; average F1 and CI are reported.

### A.2.3 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  $224 \times 224$ . Subsequently, the pretrained models are evaluated on PDAC datasets, and the corresponding test performance is presented in Figure A.4. Notably, for shot numbers of 1 and 5, our model significantly outperforms other models, demonstrating a substantial performance margin.

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  $224 \times 224$ . The pretrained models are tested on PDAC datasets, revealing similar outcomes, as illustrated in Table A.5.



pretraining method	NC	LR	RC	LR+LA	RC+LA
1-shot 5-way test					
ImageNet (FSP)	$29.57 \pm 1.07$	$31.32 \pm 1.09$	$31.16 \pm 1.07$	$30.88 \pm 1.08$	$30.14 \pm 1.08$
SimSiam	$30.48 \pm 1.08$	$30.18 \pm 1.12$	$30.19 \pm 1.13$	$30.41 \pm 1.08$	$31.13 \pm 1.10$
SimCLR	$30.79 \pm 1.08$	$30.93 \pm 1.13$	$30.78 \pm 1.12$	$31.33 \pm 1.08$	$31.22 \pm 1.07$
FSP-Patch	$34.04 \pm 1.16$	$33.99 \pm 1.20$	$33.69 \pm 1.20$	$34.29 \pm 1.15$	$34.99 \pm 1.16$
FACILE-SupCon	$35.44 \pm 1.17$	$34.94 \pm 1.20$	$34.58 \pm 1.22$	$35.68 \pm 1.16$	$35.27 \pm 1.17$
FACILE-FSP	<b><math>37.36 \pm 1.16</math></b>	<b><math>36.07 \pm 1.23</math></b>	<b><math>36.93 \pm 1.21</math></b>	<b><math>36.79 \pm 1.19</math></b>	<b><math>36.81 \pm 1.18</math></b>
5-shot 5-way test					
ImageNet (FSP)	$41.83 \pm 0.96$	$41.30 \pm 1.10$	$41.08 \pm 1.08$	$42.38 \pm 0.94$	$41.29 \pm 0.93$
SimSiam	$40.15 \pm 1.03$	$37.29 \pm 1.21$	$37.43 \pm 1.21$	$41.87 \pm 1.00$	$42.70 \pm 1.01$
SimCLR	$40.30 \pm 1.04$	$38.74 \pm 1.19$	$39.02 \pm 1.16$	$40.98 \pm 0.96$	$40.90 \pm 0.98$
FSP-Patch	$44.26 \pm 1.10$	$42.99 \pm 1.20$	$43.69 \pm 1.12$	$46.32 \pm 0.97$	$46.69 \pm 0.96$
FACILE-SupCon	$45.83 \pm 1.09$	$45.07 \pm 1.18$	$45.93 \pm 1.13$	$47.72 \pm 0.95$	$47.00 \pm 0.95$
FACILE-FSP	<b><math>48.21 \pm 1.04</math></b>	<b><math>47.62 \pm 1.12</math></b>	<b><math>47.94 \pm 1.08</math></b>	<b><math>48.84 \pm 0.95</math></b>	<b><math>48.37 \pm 0.95</math></b>

Table A.4: Models pretrained on TCGA and tested on PDAC dataset; average F1 and CI are reported.

## A.3 Datasets

### A.3.1 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.

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	<b>35.62 ± 1.10</b>	<b>35.51 ± 1.15</b>	<b>35.40 ± 1.13</b>	<b>35.87 ± 1.10</b>	<b>36.16 ± 1.09</b>
5-shot 5-way test					
SimSiam	46.00 ± 1.10	43.26 ± 1.30	44.19 ± 1.26	47.24 ± 1.00	<b>47.85 ± 1.00</b>
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	<b>46.91 ± 0.97</b>	<b>46.32 ± 1.07</b>	<b>47.10 ± 1.02</b>	<b>48.01 ± 0.90</b>	47.70 ± 0.89

Table A.5: Models pretrained on GTEx and tested on PDAC dataset; average F1 and CI are reported.

The number of slides from each organ for GTEx can be found in Figure. A.1. Thumbnails of WSI examples from the GTEx dataset can be found in Figure. A.2.

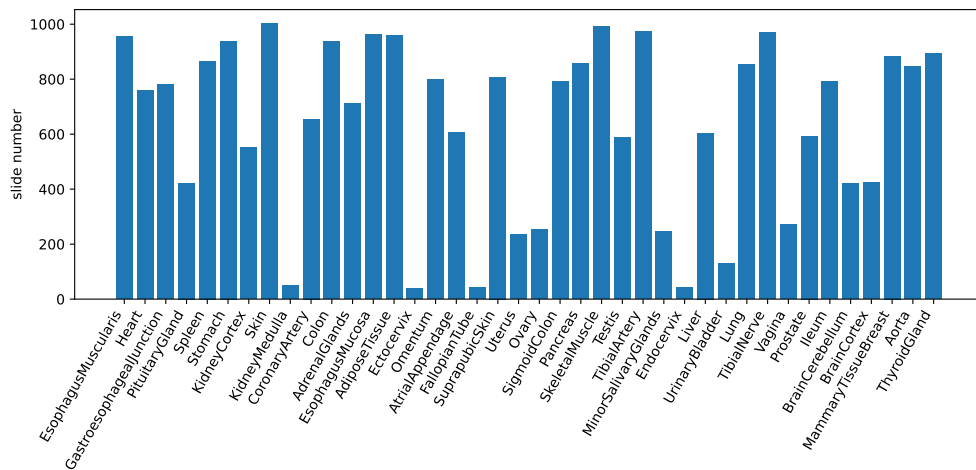


Figure A.1: Slide number for each organ in GTEx

### A.3.2 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

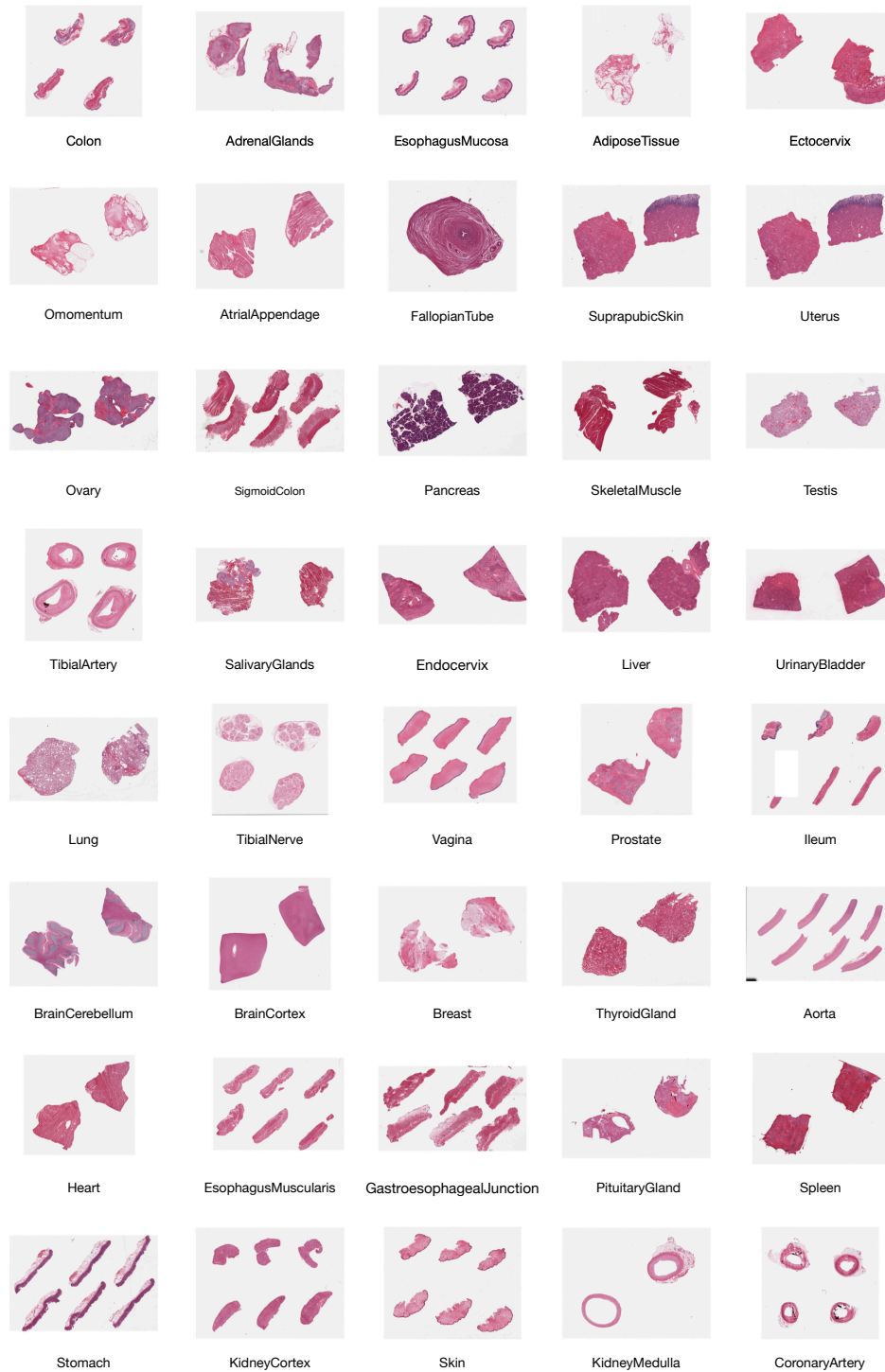


Figure A.2: Randomly deleted examples from GTEx dataset

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 the 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.,  $224 \times 224$  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. A.3. Thumb-

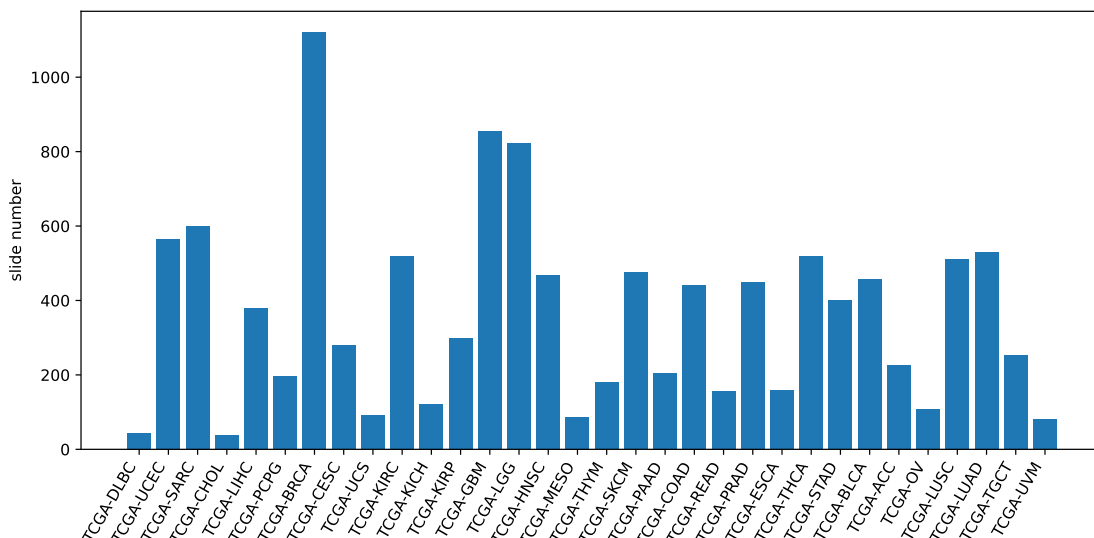


Figure A.3: Slide number for each tumor in TCGA

nails of WSI examples from TCGA dataset can be found in Figure. A.4.

### A.3.3 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



Figure A.4: Randomly selected examples from TCGA dataset

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 A.5. Furthermore, examples of patches from the PDAC dataset are provided in Figure A.6, offering visual illustrations of the dataset.

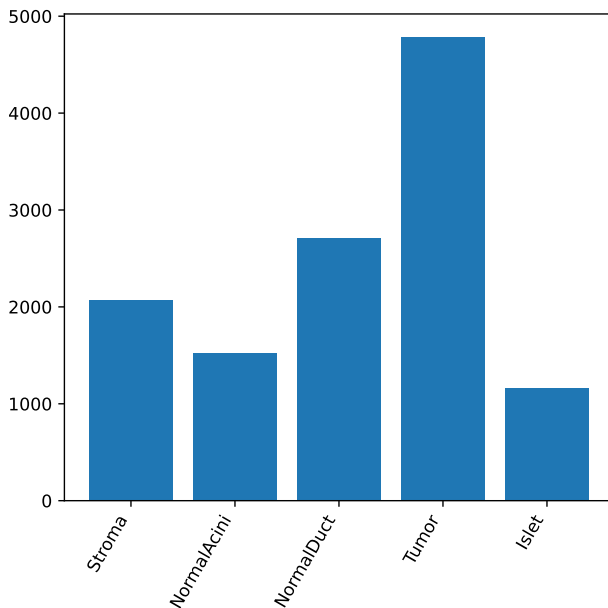


Figure A.5: Patch number for each tissue for PDAC

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 A.6: Dataset statistics

In order to validate our model on a real-world dataset, we generated WSIs of Pancreatic Ductal Adenocarcinoma (PDAC)<sup>1</sup>. PDAC, a particularly aggressive and lethal form of

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

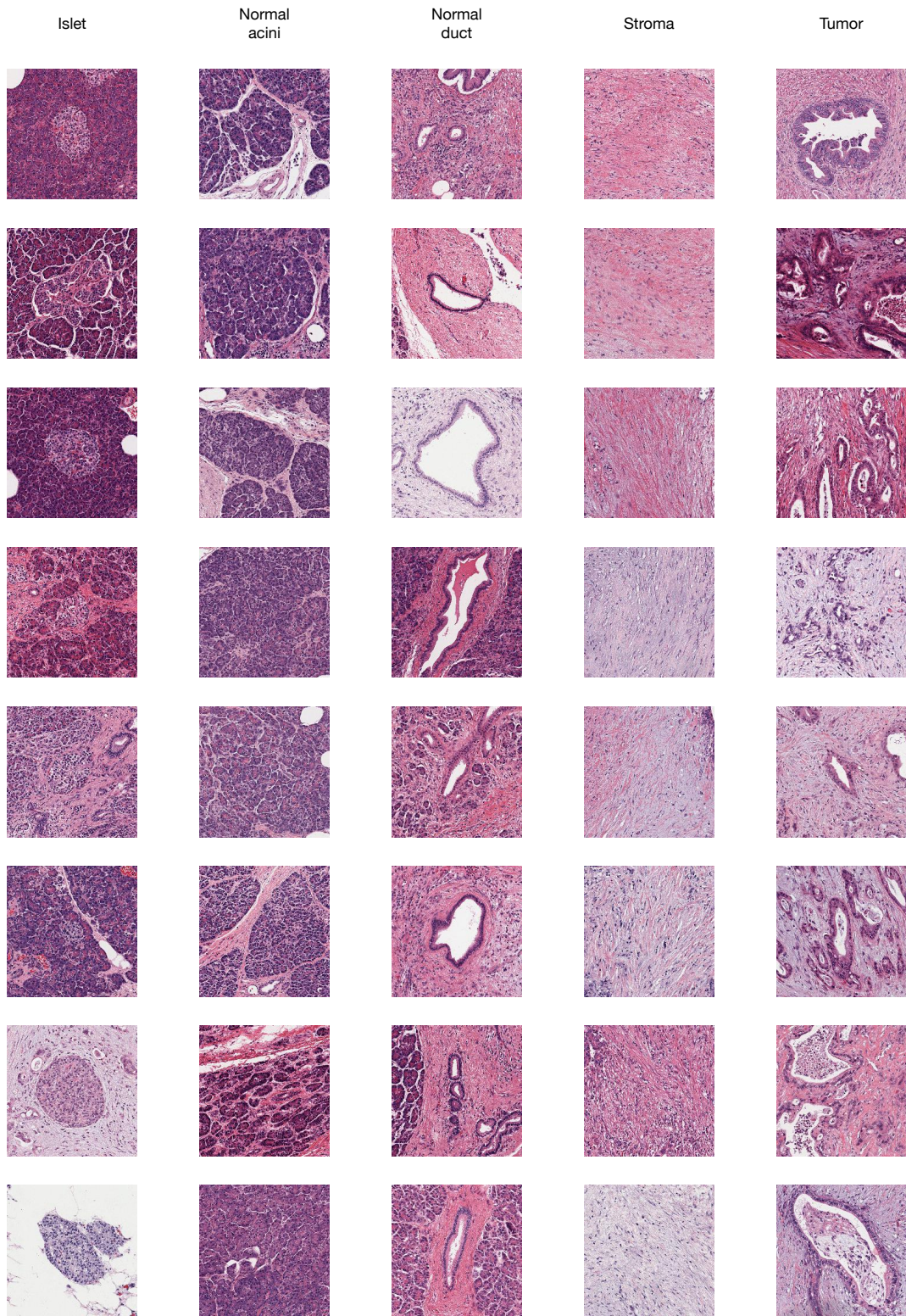


Figure A.6: Randomly selected examples from each class of PDAC dataset.

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 §A.3. There are in total 12,250 annotated patches extracted from 194 slides. The patch size used for this analysis is  $1,000 \times 1,000$  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.

#### A.3.4 *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 A.6.

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  $224 \times 224$ . 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  $768 \times 768$ . We resize the patches to  $224 \times 224$ . 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  $224 \times 224$ . 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.

## A.4 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 FSP-Patch model pretraining on the NCT dataset.

**Strong augmentation** Following previous work [Grill et al., 2020, Chen et al., 2021d, Yang et al., 2022], for SimCLR and SupCon models, we used similar strong data augmentation which contains random resized cropping, horizontal flipping, horizontal flipping, color jittering [Wu et al., 2018b] with (brightness=0.8, contrast=0.8, saturation=0.8, hue=0.2, probability=0.8), grayscale conversion [Wu et al., 2018b] with (probability=0.2), Gaussian blurring [Chen et al., 2020a] 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).

In implementing the SimSiam model, we adopted a comparable augmentation strategy, utilizing robust data augmentation techniques. Specifically, we fine-tuned parameters for color jittering, setting brightness, contrast, and saturation adjustments to 0.4, and hue to 0.1. These modifications were applied with a probability of 0.8, as informed by Chen and He [2021].

## A.5 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  $\Sigma_i$  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  $\delta$  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.

## A.6 Ablation Study

### A.6.1 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 and Storkey, 2016]. Vinyals et al. [2015a], 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 and 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.

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 and 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 the 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.

From Table A.7, 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.

### *A.6.2 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. A.7. Upon careful examination of the learning curves in Figure. A.7, we observe conclusive evidence of complete

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	<b>77.84 ± 1.16</b>	<b>77.56 ± 1.16</b>	<b>77.56 ± 1.17</b>	<b>79.16 ± 1.09</b>	<b>77.38 ± 1.18</b>
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	<b>90.91 ± 0.47</b>	<b>91.54 ± 0.46</b>	<b>91.68 ± 0.50</b>	<b>90.97 ± 0.54</b>
Set Transformer	<b>90.67 ± 0.54</b>	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	<b>52.04 ± 1.25</b>	<b>53.27 ± 1.25</b>	<b>54.19 ± 1.26</b>	<b>52.66 ± 1.25</b>	<b>52.79 ± 1.23</b>
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	<b>70.19 ± 0.87</b>	<b>70.54 ± 0.81</b>
Deep Set	<b>69.42 ± 0.85</b>	<b>69.93 ± 0.92</b>	<b>70.52 ± 0.87</b>	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	<b>68.21 ± 1.30</b>	68.17 ± 1.31	<b>68.69 ± 1.30</b>	<b>69.24 ± 1.28</b>	<b>68.18 ± 1.33</b>
Set Transformer	67.76 ± 1.31	<b>68.52 ± 1.30</b>	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	<b>87.66 ± 0.59</b>
Deep Set	85.18 ± 0.60	85.87 ± 0.60	87.11 ± 0.56	87.06 ± 0.61	85.81 ± 0.66
Set Transformer	<b>86.45 ± 0.62</b>	<b>87.74 ± 0.59</b>	<b>87.97 ± 0.58</b>	<b>88.00 ± 0.59</b>	86.92 ± 0.61

Table A.7: Performance of FACIEL-FSP with three different set-input models; average F1 and CI are reported.

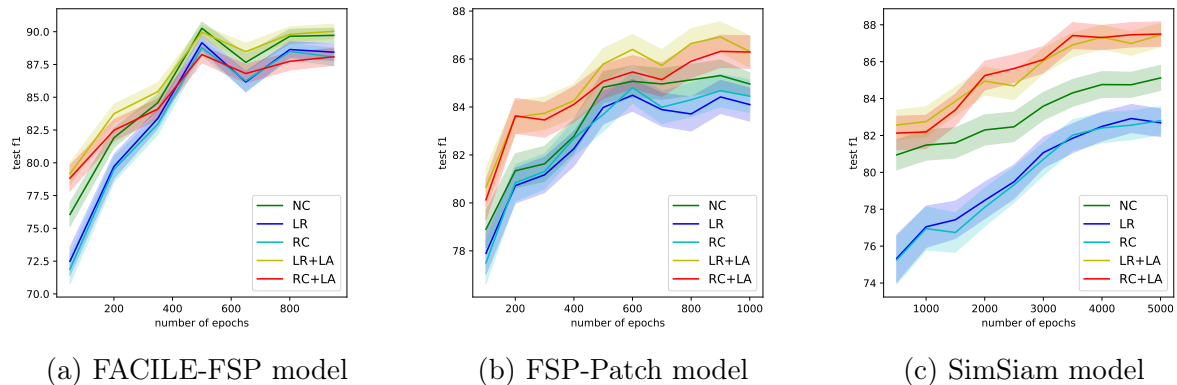


Figure A.7: Learning curves of FACILE-FSP model, FSP-Patch model, and SimSiam. The mean F1 score and CI of 5 few-shot models tested on the LC dataset with 5-shot are shown with curves.

training for all models, as they have reached convergence.

### A.6.3 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 A.8.

Observing Table A.8, 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 A.8, the best performance is attained when the input set size is 10.

## A.7 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., 2020a, Grill et al., 2020, Caron et al., 2020, Chen and 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 and Maaten [2020], Chen et al. [2020a] 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 and 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 and 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 implemen-

tation 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 §A.4.
- 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.

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  $\tilde{y}_{2k-1} = \tilde{y}_{2k} = y_k$ .

### A.7.1 SimCLR

Let  $i \in I \equiv \{1 \dots 2N\}$  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. A.8. In SimCLR, the loss takes the following form.

$$\mathcal{L}^{\text{self}} = \sum_{i \in I} \mathcal{L}_i^{\text{self}} = - \sum_{i \in I} \log \frac{\exp(z_i \cdot z_{j(i)}/\tau)}{\sum_{a \in A(i)} \exp(z_i \cdot z_a/\tau)} \quad (\text{A.1})$$

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

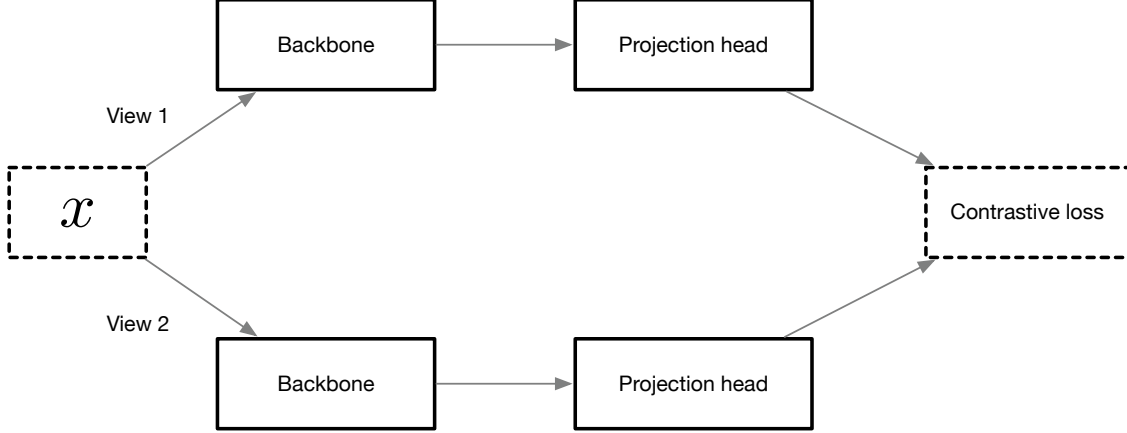


Figure A.8: Abstraction of SimCLR structure

In this paper, the  $\tau$  is always set to 0.07. The patches are augmented randomly by the augmentation module described in §A.4. We use an MLP as a projection head with two fully-connected layers, a hidden dimension of 512, and an output dimension of 512.

### A.7.2 *SupCon*

For supervised learning, the contrastive loss in Eq. (A.1) cannot handle class discrimination [Khosla et al., 2020]. Khosla et al. [2020] proposed two straightforward ways, as shown in Eq. (A.2) and Eq. (A.3), to generalize Eq. (A.1) to incorporate supervision.

$$\mathcal{L}_{\text{out}}^{\text{sup}} = \sum_{i \in I} \mathcal{L}_{\text{out},i}^{\text{sup}} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(z_i \cdot z_p / \tau)}{\sum_{a \in A(i)} \exp(z_i \cdot z_a / \tau)} \quad (\text{A.2})$$

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

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 Eq. (A.2) as the default loss. The  $\tau$  is also set to 0.07.

In our model FACILE-SupCon, 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 §A.4. 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 fully-connected layers, a hidden dimension of 512, and an output dimension of 512.

When employing set-input data with the SupCon method, the standard practice of augmenting each instance within a set poses significant challenges for the training of SupCon models. These challenges stem from two main aspects: 1) Complexity in maximizing agreement with set-input data: SupCon is traditionally trained to maximize agreement between differently augmented views of the same data point using labeled data. In our application, using set-input data means that we apply conventional data augmentation methods to each instance within a set. This results in an independently augmented set of images, as opposed to augmenting a single instance. This complexity makes it more challenging to achieve the desired maximization of agreement. 2) Constraints on batch sizes due to set inputs: Set-input models take a batch of sets as input instead of a batch of instances. It requires us to use relatively smaller batch sizes when using the same hardware configuration because of the set input. It’s important to emphasize that the batch size is a critical factor for the effectiveness of the SupCon model.

We have observed that despite these challenges, the performance of FACILE-SupCon is commendable in contexts involving smaller datasets or less complex models, i.e., CIFAR-100 in §5.3.2 and §5.3.3 or smaller trainable models as discussed in Appendices B.1 and B.3. We believe that our approach, with its nuanced application of SupCon in a set-input context, offers a valuable contribution to the field and shows the versatility of the FACILE algorithm.

### A.7.3 *SimSiam*

Non-contrastive models, e.g., SimSiam and BYOL, achieve strong results in typical ImageNet [Deng et al., 2009] pretraining setting [Chen and He, 2021, Grill et al., 2020, Li et al., 2022].



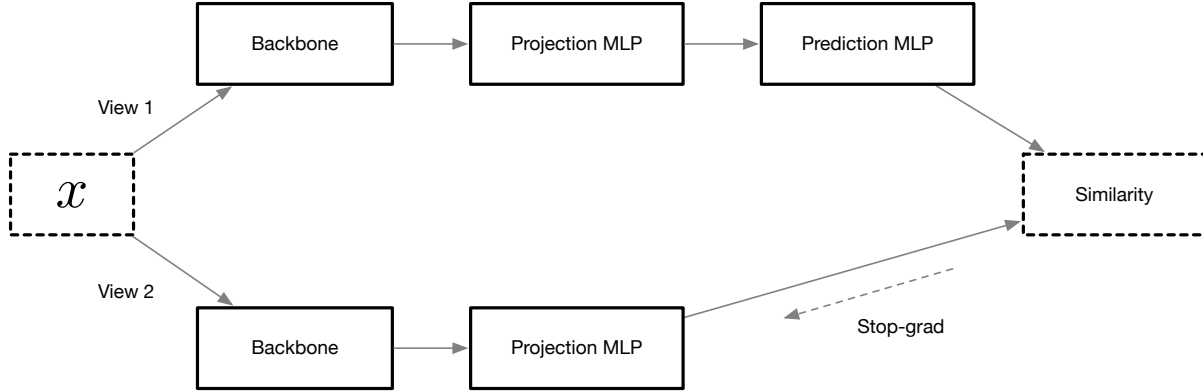


Figure A.9: Abstraction of SimSiam structure

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 and He, 2021, Li et al., 2022], which makes it extremely appealing to academics.

The abstraction of SimSiam structure is shown in Figure. A.9. 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 fully-connected layers with a hidden dimension of 512 and an output dimension of 512. It has batch normalization (BN) applied to each fully-connected layer including its output fully-connected layer. The prediction MLP also has BN applied to its hidden fully-connected layer. Its output fully-connected layer does not have BN or ReLU. The prediction MLP has 2 layers.

#### A.7.4 DINO and DINO V2

In the realm of self-supervised learning, Caron et al. [2021] introduced a novel approach that effectively utilizes concepts from knowledge distillation without the need for labels. Their proposed framework, DINO, streamlines the learning process by employing a momentum encoder within the teacher network and simplifies output prediction using standard cross-entropy loss. This method primarily depends on the centering and sharpening of outputs from the teacher network to preclude feature collapse. It notably sidelines the necessity for additional components such as predictors [Grill et al., 2020], advanced normalization techniques [Caron et al., 2020], or contrastive losses [He et al., 2020], which have been shown to contribute minimally to either stability or performance enhancements. Crucially, the DINO framework boasts flexibility, functioning effectively across both convolutional networks and Vision Transformers (ViTs) without the need for architectural modifications or specialized internal normalization adjustments [Richemond et al., 2020].

The operational mechanics of the model involve processing two distinct random transformations of a single input image through parallel student and teacher networks, which share the same architecture but differ in parameters. The output from the teacher network is first centered using the mean of the batch, and then both networks generate a  $K$ -dimensional feature vector. These vectors are normalized across the feature dimension using a temperature-controlled softmax function. The similarity between these normalized vectors is quantified using a cross-entropy loss function. To optimize learning, a stop-gradient (sg) operator is applied to the teacher network, allowing gradients to propagate exclusively through the student network. This method ensures that the teacher’s parameters are gradually updated, reflecting an exponential moving average (EMA) of the student’s parameters, thereby enhancing the overall learning efficacy and stability of the model.

Oquab et al. [2023] revisited and refined existing self-supervised pretraining methodologies, demonstrating that these approaches can generate versatile, all-purpose features when

trained on sufficiently large and diverse curated datasets. Their study enhanced pretraining scalability in terms of both data volume and model size, focusing on maximizing efficiency and stability during training. They developed an automated pipeline for assembling a dedicated, diverse, and curated image dataset, which contrasts with the typically uncurated data used in self-supervised learning. Additionally, the authors trained a ViT model with one billion parameters and successfully distilled this into smaller models that outperformed the current leading all-purpose features from OpenCLIP [Ilharco et al., 2021] across various benchmarks at both the image and pixel levels.

Oquab et al. [2023] adapted discriminative self-supervised approaches that learn features at the image and patch levels, such as those pioneered by iBOT [Zhou et al., 2021]. By reevaluating these methods with a larger dataset, they identified and implemented technical enhancements aimed at stabilizing and accelerating the learning process. These advancements not only improved the speed but also reduced the memory requirements compared to similar methods, enabling more extended training periods and larger batch sizes. This methodological evolution marks a significant step forward in developing efficient and robust models capable of handling expansive and intricate datasets in self-supervised learning.

## A.8 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., 2020a, Grill et al., 2020, Caron et al., 2020, Chen and He, 2021]. The coarse-grained data contains useful information, which is characterized by our defined Lipschitzness, to pretrain an 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*

$$\mathbb{E} \left[ \ell_{\hat{f} \circ \hat{e}}^{\text{fg}}(X, Y) - \ell_{f^* \circ e^*}^{\text{fg}}(X, Y) \right]$$

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

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

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

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

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

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

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

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  $\tilde{\mathcal{O}}(1/n)$  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  $(\ell^{\text{fg}}, \hat{P}_{Z,Y}, \mathcal{F})$ .

It is not clear whether this surrogate problem satisfies the central condition. We show that  $(\ell^{\text{fg}}, \hat{P}_{Z,Y}, \mathcal{F})$  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  $(\ell^{\text{fg}}, P_{Z,Y}, \mathcal{F})$  on  $\mathcal{Z} \times \mathcal{Y}$  is said

to satisfy the  $\epsilon$ -weak  $\eta$ -central condition if there exists an  $f^* \in \mathcal{F}$  such that

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

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,  $\ell_{f^*}^{\text{fg}}(Z, Y)$  will be close to zero most of the time.

We assume that the strong central condition holds for the learning problem  $(\ell^{\text{fg}}, P_{Z,Y}, \mathcal{F})$  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  $(\ell^{\text{fg}}, P_{Z,Y}, \mathcal{F})$  with  $(\ell^{\text{fg}}, \hat{P}_{Z,Y}, \mathcal{F})$ .

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

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

**Proof.** Note that

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

To prove that  $(\ell^{\text{fg}}, \hat{P}_{Z,Y}, \mathcal{F})$  satisfies the central condition we therefore need to bound

$\frac{1}{\eta} \log \mathbb{E}_{P_{X,Y}} \exp \left( \eta (\ell_{f^* \circ \hat{e}}^{\text{fg}} - \ell_{f \circ \hat{e}}^{\text{fg}}) \right)$  by some constant.

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

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  $\ell_{f \circ \hat{g}}^{\text{fg}} = \ell_{f \circ g_0}^{\text{fg}}$ .

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

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

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

The first inequality uses the fact that  $\ell^{\text{fg}}$  is bounded by  $B$ . The fourth line is because that  $\log x < x$ . By combining this bound with  $\epsilon$  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  $\epsilon$ -weak central condition,  $\ell^{\text{fg}}$  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  $\mathbb{E}_{Z,Y \sim Q_{Z,Y}} \left[ \ell_f^{\text{fg}}(Z, Y) - \ell_{f^*}^{\text{fg}}(Z, Y) \right]$  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  $\text{Rate}_m(\ell^{\text{cg}}, P_{S,W}, \mathcal{E}) = \mathcal{O}(\frac{1}{m^\alpha}) = \mathcal{O}(\frac{1}{n^{\alpha\beta}})$ . Proposition 7 concludes that  $(\ell^{\text{fg}}, \hat{P}_{Z,Y}, \mathcal{F})$  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  $\text{Rate}_n(\ell^{\text{fg}}, \hat{P}_{Z,Y}, \mathcal{F}) =$



$\mathcal{O}\left(\frac{d\alpha\beta \log RL'n + \log \frac{1}{\delta}}{n} + \frac{B}{n^{\alpha\beta}}\right)$ . Combining bounds with Proposition 5 we conclude that

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

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

Table A.8: Abation on set size; models tested on LC, PAIP, and NCT dataset; average F1 and CI are reported.

## APPENDIX B

### APPENDIX FOR DEEP BAYESIAN ACTIVE LEARNING

#### B.1 Implementation Details on the Empirical Example in

#### Figure. 6.1

We show an empirical example in Figure. 6.1 to provide some intuition as to why BALANCE and Batch-BALANCE are effective in practice. We train a BNN with an imbalanced MNIST training subset that contains 28 images for each digit in [1-8] and 1 image for digits 0 and 9. The cross-entropy loss is reweighted to balance the training dataset during training. We obtain 200 posterior samples of BNN and use them to get the predictions on  $\bar{\mathcal{D}}_{\text{pool}}$ . We compute the Hamming distances for predictions of all sample pairs and use these precomputed distances to plot the predictions with t-SNE [Van der Maaten and Hinton, 2008]. The equivalence classes are approximated by the farthest-first traversal algorithm (FFT) [Gonzalez, 1985].

In Figure. 6.1, the equivalence classes are highly imbalanced. The ground truth  $\bar{\mathcal{D}}_{\text{pool}}$  dataset labels represent the target hypotheses embedding. This figure highlights the scenario where the *equivalence class-based* methods, e.g. ECED and BALANCE are better than BALD.

#### B.2 Coefficient of Variation

To gain more insight into why BALANCE and Batch-BALANCE work consistently better than BALD and BatchBALD, we further investigate the dispersion of the estimated acquisition function values for those methods. Since Batch-BALANCE and BatchBALD extend their fully sequential algorithms similarly in a greedy manner, we only compare the acquisition functions of BALANCE and BALD.

The coefficient of variation (CV) is chosen for the comparison of dispersion. It is defined as the ratio of the standard deviation to the mean. CV is a standardized measure of the dispersion of a probability distribution or frequency distribution. The value of CV is independent of the unit in which it is taken.

We conduct the experiment on the imbalanced MNIST dataset in the setting of appendix B.1. We estimate the acquisition function values of BALANCE and BALD 5 times with 5 sets of  $K$  MC dropouts for each sample in the AL pool. Then, the CVs are calculated for these estimations. In Figure. B.1, we show histograms of CVs for both methods. The estimated acquisition function values of BALANCE are less dispersed, which shows potential for better performance.

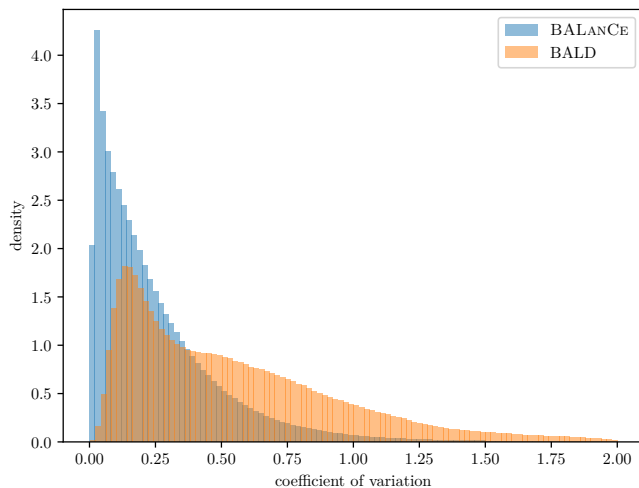
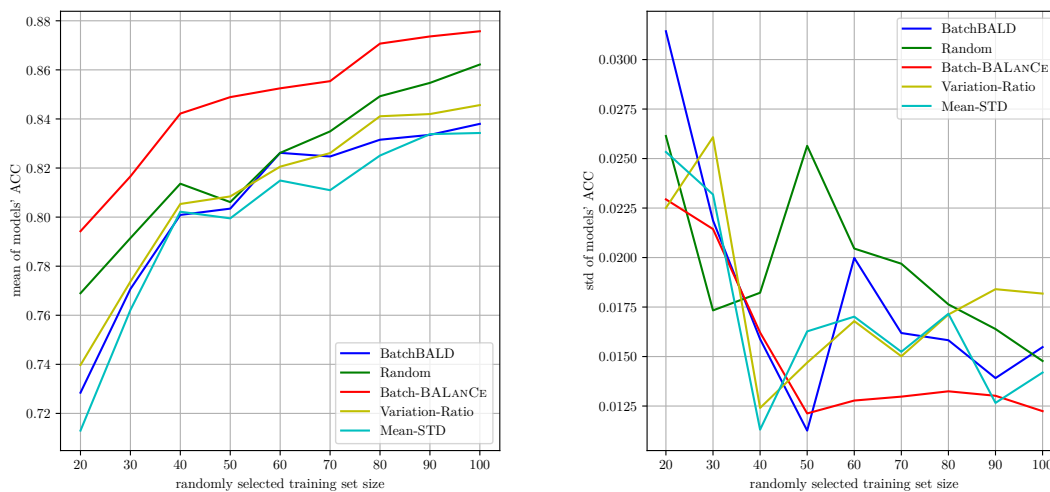


Figure B.1: Histograms for coefficient of variation.

### B.3 Predictive Variance

In order to directly compare the accuracy improvement of batches selected by different algorithms, instead of along the course of an AL trial, we conduct experiments with training sets of various sizes and compare the accuracy improvement of batches selected by AL

algorithms with the same training set. The initial training set has 10 sampled randomly from Repeated-MNIST. In each step, we select 10 random samples and add them to the training set. Hypotheses are drawn from BNN posterior given the current training set. We perform different AL algorithms and select batches with batch size 20. After each batch is added to the training set, we can estimate the accuracy improvement of the batch. In each step, we perform each AL algorithm 20 times and estimate the mean and std of accuracy improvement. The mean and std of BNNs' accuracy are shown in Figure. B.2. We can see in Figure. B.2 that our algorithms consistently select batches that have high accuracy improvement and low variance.



(a) Mean of models' accuracy on test set

(b) STD of models' accuracy on test set

Figure B.2: We empirically show AL algorithms' predictive variance.