Title
Improving Data Efficiency on Histopathology Image Analysis Using Deep Learning

Author
Li, Wenyuan

Publication Date
2020

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Improving Data Efficiency on Histopathology Image Analysis Using Deep Learning

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Electrical and Computer Engineering

by

Wenyuan Li

2020

ABSTRACT OF THE DISSERTATION

Improving Data Efficiency on Histopathology Image Analysis Using Deep Learning

by

Wenyuan Li
Doctor of Philosophy in Electrical and Computer Engineering
University of California, Los Angeles, 2020
Professor Corey Wells Arnold, Co-Chair
Professor Gregory J Pottie, Co-Chair

Ever since the advent of Alexnet in the ImageNet challenge in 2012, the medical image analysis community has taken notice of deep learning techniques and made the transition from systems that use handcrafted features to systems that learn feature from the data gradually. Histopathology images have been widely used to detect and diagnose a variety of cancers. With the growing availability of large scale gigapixel whole-slide images (WSI) of tissue specimen, digital pathology has become a very popular application area for deep learning techniques. Nevertheless, challenges exist in current computer-aided histopathology image analysis. Perhaps the biggest challenge is the insufficiency of annotated data. Deep learning requires extremely abundant training data to achieve good performance. However, only pathologists, who have been trained for years, can annotate the histopathology image accurately. Therefore, labeling histopathology images is both expensive and labor-intensive. The scarcity of the annotation can also be found at different scales. For example, to do a semantic segmentation task, it requires the network to have annotations at "pixel-wise" level; by tiling WSIs into different patches, patch-level labels are needed to provide accurate predictions. But in reality, most labels of WSIs are at case-level (*e.g.* final diagnosis) at most.

This dissertation attempts to improve data efficiency on histopathology image analy-

sis. We first start with a novel fully-supervised segmentation model for Gleason grading of prostate cancer. This method adopts two branches, an Epithelial Network Head (EHN) for detecting epithelial cells, and a Grading Network Head (GNH) for detecting, segmenting, and classifying the cancerous regions. Then we present a series of studies on semi-supervised learning, where we can take leverage of unannotated data. We focus on methods using generative adversarial networks (GANs). To this end, we demonstrate a pyramid GAN structure for high-resolution large-scale histopathology image generation and segmentation on both fully-supervised and semi-supervised scenarios. Finally, we present an active learning framework that is able to reduce the annotations required from the expert and handle noisy labels simultaneously. Extensive experiments and results have proved the effectiveness of these methods, paving the way to optimize and improve the effectiveness of data usage in histopathology image analysis.

The dissertation of Wenyuan Li is approved.

Yingnian Wu

Aydogan Ozcan

Achuta Kadambi

William F Speier

Corey Wells Arnold, Committee Co-Chair

Gregory J Pottie, Committee Co-Chair

University of California, Los Angeles

2020

Contents

List of Figures

Acknowledgments

Graduate school has been quite an experience with its ups and downs. I have learned a lot along the way and the people I met have made it a rewarding experience.

First and foremost, I would like to thank my advisor Dr. Corey Arnold for his support and guidance throughout my research. Dr. Arnold offered me the opportunity to first join the Medical Imaging Informatics (MII) group and the Computational Diagnostics (CDx) afterwards. As someone barely knowing computational diagnostics and medical image analysis before joining the lab, I learned a lot from him. I am always very thankful for all the time, efforts, and resources that he has put to my research. Besides, he also offered unconditional support for students' growth of professional career. Additionally, I would like to express my gratitude to Dr. William Speier, who spent countless hours discussing the project with me and inspiring me to do more rigorous works. I would also like to offer my sincere thanks and respect to the co-chair of my committee, Dr. Greg Pottie, for his insights and dedication to help me grow my knowledge. I would like to thank all the other members of my thesis committee, Dr. Yingnian Wu, Dr. Aydogan Ozcan, and Dr. Achuta Kadambi, for offering guidance and feedback for all stages of this dissertation work.

I am also grateful that I had the opportunity to work with Mrs Jiayuan Li, without whom I would not have been able to complete my PhD research. She offered tremendous help to me ever since I joined the group. I really enjoyed the time to work with her.

To all current and past members of the MII and CDx group, thank you for providing me an intellectually simulating environment for my study and research. These people are Panayiotis Petousis, Nova Smedley, Karthik Sarma, Simon Han, Yiwen Meng, Tianran Zhang, Jennifer Polson, Harry Zhang, Zichen Wang, Alex Raman, Shiwen Shen, Nicholas Matiasz, Edgar Rios, Daniel Johnson, Leihao Wei, and Yannan Lin. Thank you to the professors, Dr. Alex Bui, Dr. Denise Aberle, Dr. William Hsu, and Dr. Ricky Taira, for their kindness, generosity, professionalism, and mentoring me in various research topics. Thank you to the stuff members, Isabel Rippy, Lew Andrada, Shawn Chen, Patrick Langdon, and Denise

Luna, who always took care of administrative matters for the students professionally.

I would like to thank NIH, the UCLA Radiology Department Exploratory Research Grant Program, the UCLA Electrical and Computer Engineering Program, and Chiang Chen Industrial Charity Foundation for funding my study and research.

Last but not the least, I would like to thank my parents Fengxian Zhang and Xingwei Li, and my beloved one Dongyu Chen for being unconditionally supportive and encouraging during graduate studies.

Curriculum Vitae

2010 – 2014	B.E. in Optical Engineering, Zhejiang University, Hangzhou, P.R.China.
2014 – 2016	M.S. in Electrical Engineering, UCLA, Los Angeles, USA.
2016 – 2020	Graduate Student Researcher at Electrical and Computer Engineering, UCLA, Los Angeles, USA.
2018	Research intern at IQVIA.
2019	Software engineer intern at Facebook.

Selected Publications

[1] Wenyuan Li, Zichen Wang, Yuguang Yue, Jiayun Li, William Speier, Mingyuan Zhou, and Corey Arnold. Semi-supervised learning using adversarial training with good and bad samples. *Machine Vision and Applications*, 31(6), 1-11, 2020 .

[2] Wenyuan Li, Zichen Wang, Jiayun Li, Jennifer Polson, William Speier, and Corey Arnold. Semi-supervised learning based on generative adversarial network: a comparison between good GAN and bad GAN approach. *CVPR Workshops*, 2019 .

[3] Jiayun Li, Wenyuan Li, Arkadiusz Gertych, Beatrice S Knudsen, William Speier, and Corey Arnold. An attention-based multi-resolution model for prostate whole slide image classification and localization *CVPR Workshops*, 2019 .

[4] Wenyuan Li, Jiayun Li, Karthik V Sarma, King Chung Ho, Shiwen Shen, Beatrice S Knudsen, Arkadiusz Gertych, and Corey W Arnold. Path r-cnn for prostate cancer diagnosis and gleason grading of histological images. *IEEE transactions on medical imaging*, 38(4):945–954, 2018 .

[5] Wenyuan Li, Yunlong Wang, Yong Cai, Emily Zhao, Yilian Yuan, and Corey Arnold. Semi-supervised rare disease detection using generative adversarial network. *NeurIPS Workshops*, 2018 .

CHAPTER 1

Introduction

1.1 Motivation

Deep neural networks (DNNs) have rapidly become a methodology of choice for analyzing medical images in recent years [88]. One of the most prominent methods of DNNs that has achieved tremendous success in image analysis is convolutional neural networks (CNNs). CNNs contain many layers that transform input images to the output target with convolution filters. Ever since the most recent CNNs' breakthrough by Krizhevsky *et al.* to the ImageNet challenge in 2012 [70], the medical image analysis community has taken notice of these pivotal developments and gradually transitioned from systems that use handcrafted features to systems that learn features from the data. DNNs/CNNs have been used for image classification, object detection, semantic segmentation, image registration, and other tasks in the medical domain.

The growing availability of large scale gigapixel whole-slide images (WSIs) of tissue specimen has made digital pathology a very popular application area for DNNs/ CNNs. The most actively researched task in digital pathology image analysis is computer-assisted diagnosis (CAD), where the computer algorithm is used to help the expert to predict the final diagnostic outcome. In this regard, deep learning techniques have been applied for detecting, segmenting or classifying nuclei, large organs or disease severity in different cases. Since the errors made by a machine learning system reportedly differ from those made by a human pathologist [133], the diagnostic accuracy could be improved using a CAD system. CAD may also reduce the variability in interpretations and prevent overlooking by investigating all pixels within WSIs [65]. Additionally, recent works have applied deep learning techniques

for image normalization [115], image re-staining [56], content based image retrieval [125], *etc.*, of digital histopathology.

1.2 Challenges and Objectives

Though it has become easier and cheaper to get the digitized histopathology images, there are some unique characteristics of histopathology image analysis and computational challenges to treat them.

Insufficient Labeled Images

Probably the biggest problem in histopathology image analysis using DNNs is that only a small number of labeled data is available. The success of deep learning techniques requires extremely abundant training data. Annotations for the natural image analysis can be easily retrieved from the internet and it is also possible to use crowd-sourcing approach since anyone can identify simple objects such as "cat" and "dog". However, only pathologists, who have been trained for years, can annotate the histopathology images accurately. Therefore, labeling histopathology images is both expensive and labor-intensive. The scarcity of the annotation can also be found at different scales. For example, to do a semantic segmentation task, it requires the network to have annotations at "pixel-wise" level; by tiling WSIs into different patches, patch-level labels are needed to provide accurate predictions. Nevertheless, most labels of WSIs are at case-level (*e.g.* final diagnosis) at most.

As reflected by the title *Improving Data Efficiency on Histopathology Image Analysis using Deep Learning*, throughout the dissertation we are trying to address the label scarcity challenge. We start from the fully supervised learning semantic segmentation task in Chapter 2, and gradually move to semi-supervised learning that can take leverage of the unlabeled data in Chapter 3, Chapter 4, and Chapter 5. Finally, we demonstrate an active learning framework that is able to identify the most "infromative" data for annotations and lead to the labeling effort reduction in Chapter 6.

Very Large Image Size

DNNs were first applied to relatively smaller image sizes, such as 256×256 pixels, and achieved good success. Since increasing the size of the input image results in increasing the number of parameters to be estimated, and the required computational memory and power, images with large size often need to be scaled into smaller size that still permit sufficient distinction. Unfortunately, rescaling WSIs for analysis may not be as successful as it is for natural images. WSIs usually have gigapixels, which contain complex structures such as cells and glands. Information regarding cellular level features such as cells shape are well captured in high-power field microscopic images, but structural information such as a glandular structure made of many cells are better captured in a lower-power field. Both of these features are needed for an accurate clinical diagnosis. Simply rescaling the entire image to a smaller size, such as 256×256 pixels, would lead to the loss of information at the cellular level, resulting in the decrease of the diagnostic accuracy.

To solve this problem, entire WSIs are commonly tiled into partial regions of smaller patches (*e.g.* 256×256), and each patch is analyzed independently. More sophisticated algorithms can be developed during this process. For example, in Chapter 2, we develop a region-based CNN to first detect the region of interests (ROIs) and then zoom in the area for detailed analysis; once the analysis is done for each single patch, we stitch all of them back and apply conditional random field to remove the unnatural predictive boundary between patches. In Chapter 5, we propose a novel high-resolution large-scale histopathology image generation and segmentation framework by hierarchical structures to enable the image analysis with large size on higher-power field.

Low Concordance Rate and Noisy Labels

Besides being time-consuming and labor-intensive, manually annotating the histopathology images can also be plagued by inter- and intra-observer variability. This problem is particularly pronounced when differentiating the hard cases (*e.g.* Gleason 3 (G3) vs. Gleason 4

(G4) in prostate cancer). In a Gleason grade study for prostate cancer, the concordance rate of multiple pathologists can be as low as 57.9% [144]. This fact will make the annotations inevitably noisy. At the same time, it is easy for the DNNs with huge capacity to fit noisy annotations, which can hurt their generalization ability for the real clinical usage. Furthermore, it is challenging to distinguish mislabeled samples from hard samples. Mislabeled samples are samples with wrong annotations, while hard samples have the right label but the samples themselves are not "typical". The lack of massive and clean annotations are big challenges in histopathology image analysis. They make the capability of DNNs unscalable to the size of collected data.

A CAD tool could impact clinical practice by providing a repeatable and more precise method for diagnosis. Compared with the traditional method, an automated analysis system would alleviate the inter- and intra-observer variability from the pathologists. Furthermore, we propose a machine learning methodology to distinguish between noisy samples and hard samples in Chapter 6. By excluding noisy samples, we prevent them from hurting the DNNs performance.

1.3 Contributions and Novelties of Dissertation

The contributions and novelties of the dissertation are summarized as follows.

A Region-based CNN for Gleason Grading of Prostate Cancer

We start with a fully supervised region-based convolutional neural network (R-CNN) for Gleason grading of prostate cancer. Prostate cancer is the most common and second most deadly form of cancer in men in the United States. The classification of prostate cancers based on Gleason grading using histological images is important in risk assessment and treatment planning for patients. Here, we demonstrate our R-CNN for multi-task prediction using a Epithelial Network Head and a Grading Network Head. Compared to a single task model, our multi-task model can provide complementary contextual information, which contributes

to better performance. Our model achieved state-of-the-art performance in epithelial cells detection and Gleason grading tasks simultaneously. Using five-fold cross-validation, our model achieved an epithelial cells detection accuracy of 99.07% with an average AUC of 0.998. As for Gleason grading, our model obtained a mean intersection over union of 79.56% and an overall pixel accuracy of 89.40%.

The main contributions of our work are twofold: first, by adding an Epithelial Network Head (EHN), we adapted the Mask R-CNN to be suitable for the histological image analysis for Gleason grading task with little additional computational overhead; second we developed a two-stage training strategy which enables our model to detect epithelial cells and predict Gleason grades simultaneously.

Semi-supervised Learning Framework using Generative Adversarial Network

We explore the potential usage of generative adversarial network (GAN) in semi-supervised learning for histopathology images. Specifically, we study semi-supervised semantic segmentation in Chapter 5. Semantic segmentation of histopathology images can be a vital aspect of computer-aided diagnosis, and deep learning models have been effectively applied to this task with varying levels of success. However, their impact has been limited due to the small size of fully annotated datasets. Data augmentation is one avenue to address this limitation. Generative Adversarial Networks (GANs) have shown promise in this respect, but previous work has focused mostly on classification tasks applied to MR and CT images, both of which have lower resolution and scale than histopathology images. There is limited research that applies GANs as a data augmentation approach for large-scale image semantic segmentation, which requires high-quality image-mask pairs. In this work, we propose a multi-scale conditional GAN for high-resolution, large-scale histopathology image generation and segmentation. Our model consists of a pyramid of GAN structures, each responsible for generating and segmenting images at a different scale. Using semantic masks, the generative component of our model is able to synthesize histopathology images that are visually realistic. We demonstrate that these synthesized images along with their masks can be used

to boost segmentation performance especially in semi-supervised scenarios.

The main contributions of this work are twofold. First, by using a pyramid generation scheme, we are able to generate large-scale histopathological images up to 1024×1024 at high resolution (20x). Compared to the state-of-the-art pathology synthesis methods, which generate images up to 256×256 allowing for only limited context such as simple nuclei ([93, 114]), our generation allows us to incorporate richer context such as gland structures and nuclei details that are useful for precise diagnosis. Second, the generation is based upon a conditional method, which produces good image-mask pairs. These image-mask pairs can be used to compensate for the lack of data points in training segmentation models. We demonstrate the effectiveness of our method in segmentation tasks and analyze how it performs differently in supervised and semi-supervised settings.

An Active Learning Framework for Histopathology Image Analysis

We develop an active learning framework that is tailored to histopathology image analysis, namely PathAL, in Chapter 6. PathAL is able to dynamically identify the noisy labels and sample the images that need to be annotated. We provide a solution that is able to reduce the annotations required from the expert and handle noisy labels simultaneously. Specifically, for each iteration of PathAL, we first train the network using the annotated images. Then we make the network to transfer from overfitting to underfitting status cyclically by adjusting the hyper-parameters. In this process, we monitor and rank the normalized average loss of every labeled example and the normalized average prediction entropy of every unlabeled example. We also measure the complexity of data points using their distribution density in the feature space, and rank their complexity in an unsupervised manner. By doing so, the noisy labeled samples can be identified and discarded, while the hard and minority samples can be preserved; the unlabeled images that are most informative to the model as it trains are selected for annotations and added to the training for the next iteration. In addition, the typical unlabeled samples with highest predictive confidence are added to the training pool with pseudo annotations generated by the model itself. This cost-effective sample selection

strategy is able to improve the classification performance with much less manual annotations. Our proposed method is a tailor-made strategy for histopathology image analysis. The main contributions of this work include: 1) an active learning framework (PathAL) that is able to dynamically identify important samples to annotate, distinguish noisy and hard samples in the training sets, is proposed; 2) extensive experiments are done to show promising results on enhancing the model performance with much less annotation efforts and noisy samples.

1.4 Organization of Dissertation

The remainder of the dissertation is organized as follows. In Chapter 2, we present region-based CNN for Gleason grading of prostate cancer. This chapter is based on my previous publication [82]. In Chapter 3, we discuss our initial exploration on semi-supervised learning using a generative adversarial network (GAN). Further, we present UGAN, a semi-supervised method that uses both good and bad samples in Chapter 4. In Chapter 5, we develop a hierarchical model for high-resolution large-scale histopathology image generation and segmentation. These chapters are based on my previous publications [84,84] and a prepared manuscript. In Chapter 6, an active learning framework that is tailored to histopathology images is presented. This chapter is based a manuscript in preparation for submission. Chapter 7 concludes the dissertation and highlight open research questions.

CHAPTER 2

Path R-CNN for Prostate Cancer Diagnosis and Gleason Grading of Histological Images

2.1 Introduction

Supervised learning (SL) attempts to learn a function that maps an input to an output based on a set of training examples. In supervised learning, each example is a pair consisting of an input object and a desired output value, *i.e.* SL requires each training sample to be paired with a human annotated label. In this section, we will first introduce our supervised learning effort, namely "Path R-CNN", on the prostate cancer Gleason grading task.

2.1.1 Motivation

Prostate cancer is the most prevalent form of cancer and the second deadliest cancer in men in the U.S. [118]. Pathologists use several screening methodologies to qualitatively describe the diverse tumor histology in the prostate. Normal prostate tissue includes stroma and glands. Stroma is the fibromuscular tissue surrounding glands. Each gland unit is composed of a lumen and rows of epithelial cells located in an orderly fashion around it. The stroma holds the gland units together. Cancerous tissue has epithelial cells that replicate in an uncontrolled manner, disrupting the regular arrangement of gland units. In high grade cancer, both stroma and lumen are generally replaced by epithelial cells.

One of the most reliable methods to quantify prostate cancer aggressiveness is through the Gleason grading system [34]. Gleason grades are used to describe growth patterns in prostate adenocarcinoma and are related to severity of disease. Gleason grades range from

Gleason 1 (G1) to Gleason 5 (G5), with a score of G1 corresponding to tissue with the highest degree of resemblance to normal tissue and best prognosis, and a score of G5 corresponding to poorly differentiated tissue and the poorest prognosis.

The Gleason grading system continues to be updated by the consensus of the International Society for Urological Pathology [25]. This system supports clinical decision-making in a number of ways. First, the grades help physicians identify the extent of the disease. Second, the grades correlate well with patient outcomes. Finally, the grades aid in the determination of the most appropriate treatment options [24].

However, to date, most Gleason scores are assigned manually through pathologist review, a process that is time-consuming and plagued by inter- and intra-observer variability. This problem is particularly pronounced when differentiating Gleason 3 (G3) vs. Gleason 4 (G4), a distinction that may have substantial impact on further treatment [49, 51, 74].

Therefore, a CAD tool for Gleason grading could impact clinical practice by providing a repeatable and more precise method for grading prostate cancers. Compared with traditional methods, an automated Gleason grading system would alleviate a time-consuming portion of the pathologist's workload.

2.1.2 Related Works

In this section, we review the related works from the literature from three perspectives. First, we briefly review the previous CAD work on prostate cancer diagnosis. Then, several recent representative biomedical image segmentation methods are discussed. Finally, we review the region-based convolutional neural networks (R-CNN) approach for object detection and instance segmentation [44], upon which our proposed method is based.

2.1.2.1 Prostate Cancer Diagnosis and Gleason Grading of Histological Images

A few previous papers have been published in developing an automatic Gleason grading system for prostate cancer diagnosis. A commonly used approach is to extract tissue features

and apply classifiers upon the selected features. Stotzka *et al*. [128] extracted statistical and structural features from the spatial distribution of epithelial nuclei over the image area. They used a hybrid neural network/Gaussian statistical classifier to distinguish moderately and poorly differentiated histological samples. Smith *et al*. [123] used the power spectrum of tissue images to represent their texture characteristics. They used a nearest neighbor classifier to assign the input image to Gleason grades 1 through 3 and the combined grades of 4 and 5. Wetzel *et al*. [139] proposed the use of features derived from spanning trees connecting cell nuclei across the tumor image to represent tissue images belonging to each grade. Jafari-Khouzani and Soltanian-Zadeh [55] used features based on co-occurrence matrices, wavelet packets, and multi-wavelets combined with a k-nearest neighbor (kNN) classifier to classify each image into grades 2 through 5. Farjam *et al*. [26] proposed a multistage classifier based on morphometric and texture features for Gleason grading. First, gland units are identified using texture features. Then, morphometric and texture features obtained from gland units are used in a series of classification stages to classify the image into grades 1 through 5. Tabesh *et al*. [130] aggregated color, texture, and morphometric cues at the global and histological object levels for classification and compared Gaussian, k-nearest neighbor, and support vector machine classifiers along with the sequential forward feature selection algorithm. Nguyen *et al*. [100] used structural features of prostate glands to classify pre-extracted regions of interest (ROIs) into benign, G3, and G4. Gorelick *et al*. [38] proposed a two stage Adaboost model to classify around 991 sub-images extracted from 50 whole-mount sections of 15 patients.

Though most of these papers achieved good results on their datasets due to heavy reliance on feature extraction, the systems described above are prone to subjectivity and limited intra- and inter-system reproducibility. Moreover, all of the systems require accurate localization of the small image area (region of interest, RoI) to extract features from, which is a non-trivial problem [22].

2.1.2.2 Deep Learning Models for Biomedical Image Segmentation

Recent developments using deep convolutional neural networks (CNNs) [76], particularly fully convolutional networks (FCNs) [89], have demonstrated success for biomedical image analysis [5, 6, 40, 57, 88]. These neural network approaches learn features directly, rather than using handcrafted features. Ronneberger *et al.* [111] proposed U-Net, a U-shaped neural network that consists of a contracting path to capture context and a symmetric expanding path that enables precise localization. The Multi-scale U-Net proposed by Li *et al.* [79] incorporated different scale input information without overly increasing memory requirements and achieves better results than the original U-Net and the previous work by Gertych *et al.* [30]. A more comprehensive comparison was done by Ing *et al.* [52], where they tested four CNNs including FCN-8s, two SegNet variants, and multi-scale U-Net for performance in semantic segmentation of high and low Gleason grade tumors. Chen *et al.* [13] proposed DCAN, which added a unified multi-task object to the U-Net learning framework, which won the MICCAI2015 Gland Segmentation Challenge [120]. Based on DCAN, Yang *et al.* [146] proposed suggestive annotation, which extracts representative samples as a training dataset, by adopting active learning into their network design. With the refined training sample and optimized structure, suggestive annotation achieves state-of-the-art performance on the MICCAI Gland Segmentation dataset [120]. More recently, Li *et al.* [81] have proposed a semi-supervised learning method using the expectation maximization in a deep learning framework for prostate cancer grading. The successes of the above methods demonstrate that deep learning has substantial applicability to medical image analysis. Moreover, multi-task learning that provides more information to train the network [13], and deep active learning [146] that helps the model focus on representative images, have both been proven to boost performance. In the same vein, we have developed a model that adopts an R-CNN into a larger framework.

2.1.2.3 R-CNN Approach on Image Segmentation

Object proposal methods were first adopted in CNNs [70] by R-CNN [33]. The R-CNN method trains CNNs end-to-end to classify the proposed RoIs into object categories or background. Fast R-CNN [32] advanced R-CNN to allow extracting RoIs on feature maps using an *RoIPool* layer, improving both speed and accuracy. Faster-RCNN [110] followed this path and extended it by learning an attention mechanism with a Region Proposal Network (RPN), which simultaneously predicts object bounds and objectness scores at each position. The uniqueness of these R-CNN methods is that by using RPN components, the network learns where to focus within a given image.

Driven by the success of R-CNN and its extensions, many recent approaches to image segmentation are based on *segment proposals*. In particular, Mask R-CNN [44] added a third branch that outputted the object mask on the basis of Faster R-CNN [110] and demonstrated remarkable power on image instance segmentation. In their network settings, segmentation masks were generated for every class without competition among classes, while relying on the classification branch to predict the class label. This is different from previous deep-learning based segmentation methods [79, 89, 111] where classification and segmentation tasks were coupled by a pixel-wise soft-max layer. This difference is the key for the improved instance segmentation results. In addition, Mask R-CNN proposes a "RoIAlign" layer, that faithfully preserves exact spatial locations. The "RoIAlign" layer properly aligns the extracted features from the network with the input image, which improves segmentation accuracy by a large margin. However, the "RoIAlign" layer extracts features for each RoI at the same scale; this works well for natural image instance segmentation but might not be effective for medical image analysis as we will discuss in Section 2.4. We refer readers to [44] for more details of Mask R-CNN.

2.1.3 Contributions

In this chapter, we propose a novel model that can automatically diagnose prostate cancer and perform Gleason grading based on histological whole slide images. Compared with previous work, our proposed method achieves state-of-the-art performance in both epithelial cells detection and Gleason grading accuracy. The main contributions of our work are twofold: first, by adding an Epithelial Network Head (EHN), we adapted the Mask R-CNN to be suitable for the histological image analysis for Gleason grading task with little additional computational overhead; second we developed a two-stage training strategy which enables our model to detect epithelial cells and predict Gleason grades simultaneously. Extensive experimental results show that our model achieved state-of-the-art performance in epithelial cells detection and Gleason grading tasks simultaneously. Using five-fold cross-validation, our model achieved an epithelial cells detection accuracy of 99.07% with an average AUC of 0.998. As for Gleason grading, our model obtained a mean intersection over union of 79.56% and an overall pixel accuracy of 89.40%.

2.1.4 Organization

The rest of the chapter is organized as follows. We start with a brief introduction of our dataset and proposed method are described in Section 2.2. In Section 2.3 we present our experimental results. We then discuss the limitations of our work and provide directions for possible future work in Section 2.4. Finally, conclusions are drawn in Section 2.5, which concludes the chapter.

2.2 Methods

In this section, we first describe the dataset we used for our effort. After that, we formally define our problem in the context of image instance segmentation problem. Then, we describe the novel framework that we used to solve our problem in detail. Finally, we provide evaluation metrics on which our model was assessed and compared with previous efforts.

Table 2.1: Dataset summary.

	No. Image	No. Patient	Label Set
SetA [30]	224	20	Stroma, Benign, Low-grade (CG3), High-grade (CG4)
SetB [52]	289	20	Stroma, Benign, Low-grade (CG3), High-grade (CG4, CG5)
Total No. Image: 513		Total No. Patient: 40	

2.2.1 Dataset

Our dataset consists of 513 images, which were retrieved from archives in the Pathology Department at Cedars-Sinai Medical Center (IRB# Pro00029960). The 513 images are combined from two sets of tiles. 224 of the images are from 20 patients and contain stroma (ST), benign or normal glands (BN, rated as GG2 or below), low-grade cancer (LG, image areas rated as GG3) and high-grade cancer (HG, image areas rated as GG4) (**Set *A***) [30]. The remaining 289 images are from 20 different patients and contain dense high-grade tumors including Gleason grade 5 (GG5) as well as Gleason grade 4 (GG4) with cribriform and non-cribriform glands. In addition, some of these images contain only stromal constituents such as nerve tissue and blood vessels (**Set *B***) [52]. Slides from **Set *A*** were digitized using a high resolution whole slide scanner SCN400F (Leica Biosystems, Buffalo Grove, IL), whereas slides from the **Set *B*** were acquired through the Aperio scanning system (Aperio ePathology Solutions, Vista, CA). The scanning objective in both systems was set to 20x. The output was a color RGB image with the pixel size of 0.5 $\mu m \times 0.5$ μm and 8 bit intensity depth for each color channel. Representative tiles previously identified by the pathologist were extracted from whole slide images (WSIs) and then saved as 1200×1200 pixel tiles for analysis. The content of each tile was hand-annotated by an expert research pathologist using an in house developed graphical user interface [8, 27, 30]. Figure 2.1 shows three representative examples

from the dataset we used in this study. All annotated image tiles were cross-evaluated by the pathologists, and corrections made by consensus. All tiles were normalized to account for stain variability in the pre-processing stage [109]. Data augmentation including, image flip, mirror, and rotate, were applied to the tiles before being fed into the network. These two datasets were also used in previous studies in [30] and [52]. For more information about the Gleason grading system and how we classify the tissues into four categories, we refer readers to the Appendix Appendix A.1.

2.2.2 Problem Definition

Here, we formulate the prostate cancer diagnosis and Gleason grading problem in the context of a common computer vision problem, instance segmentation. We assigned the stromal components of the input images as the background class. Other epithelial cells in the input image that have been annotated by the pathologists as benign, low-grade or high-grade were assigned as instance objects, *i.e.* the RoIs we want our network to find. Under these assignments, the epithelial detection is a natural binary classification problem, in which our network needs to output 1 if there are any specific RoIs in the image or 0 if the whole input image contains only stroma. The Gleason grading problem involves *detection* of the epithelial cells' areas, *classification* of the grade of each area, and *segmentation* of the epithelial areas from the background. These questions can be solved by object detection (draw a bounding box around the epithelial cells' areas), object classification (classify each epithelial cell's area into different categories: benign, low-grade, *etc.*), and instance segmentation (draw a segmentation mask for each epithelial area). The right column of Figure 2.1 demonstrates this idea. Each epithelial area (RoI) is represented by a unique color, which has a bounding box, class label and segmented mask associated with it.

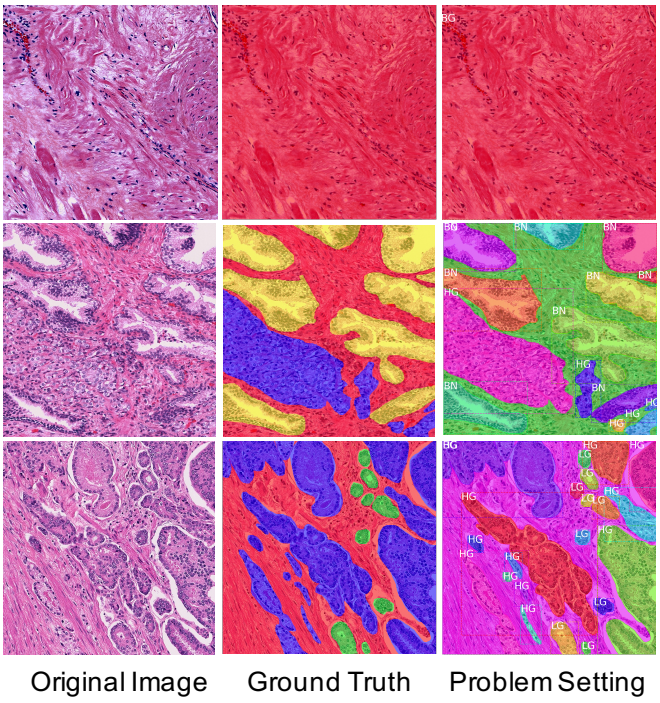

Figure 2.1: Samples from the dataset used for this work. Three representative examples are shown. The top row shows a stroma-only example; the middle row is an example with a large benign region; the bottom row is an example with both high-grade and low-grade cancer. **(Left Column)**: Original histological image tiles stained by H&E. **(Middle Column)**: Micrographs annotated by pathologists for stroma (red), benign glands (yellow), low-grade cancer (green), and high-grade cancer (blue). **(Right Column)**: Annotated data used to form a multi-task problem. We treat stroma as background (BG), and each cancer area as a separate object with a bounding box, class label, and segmented mask as its properties (BN: benign, LG: low-grade, HG: high-grade).

2.2.3 Model Definition

2.2.3.1 Network Architecture

Figure 2.2 shows the entire system and the components of the proposed model. We use ResNet as the backbone for our image parser. First, the image parser generates feature maps. These feature maps are then fed into two branches. In the left branch, we adopted the same two-stage procedure as in the Mask R-CNN. The feature maps are first used by a Region Proposal Network (RPN) that generates region proposals (RoIs). In the second stage, a Grading Network Head (GNH) is then used for predicting the class, box offset, and a binary mask for each RoI. To this we add a right branch that outputs an epithelial cell score that detects the presence of epithelial cells in the image. We refer to this part as the Epithelial Network Head (ENH). The final prediction of the network depends on the results of the ENH and GNH. Finally, a post-processing step based on a conditional random field is applied to the prediction. Because our model is inspired by Mask R-CNN [44], we name it Path R-CNN.

2.2.3.2 Objective Function

The goals of our model are to detect the presence of epithelial cells and to output a Gleason grade segmentation mask. The ENH and GNH are designed to complete these two tasks separately. In the GNH, there are three separate networks. We define classification loss L_{cls}, which evaluates whether the model can output Gleason grades accurately, bounding-box loss L_{cls}, which evaluates whether the model can locate the epithelial cells accurately, and mask loss L_{mask}, which evaluates whether the model can segment the epithelial regions' boundaries accurately. The objective function for training the model follows the same spirit in Mask R-CNN [44] and Faster R-CNN [110] that applies bounding-box classification, regression and per-pixel sigmoid mask segmentation. In addition, we add an objectness prediction loss L_{obj} for the ENH, which represents misclassification of whether there are epithelial cells in the given pathological image. L_{obj} is designed as a common binary classification loss, which is

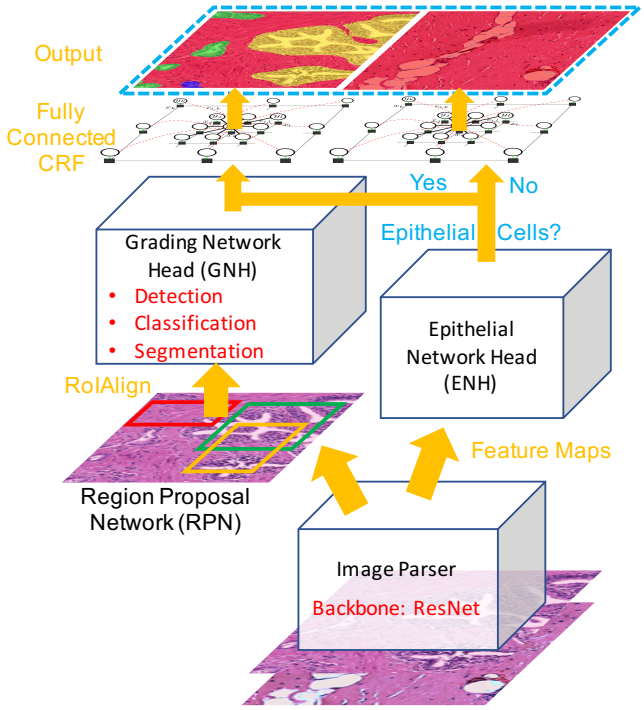

Figure 2.2: Overview of the proposed Path R-CNN model architecture. We use the ResNet model as a backbone to extract feature maps from the input image. Extracted feature maps are then fed into two branches. In the left branch, the region proposal network (RPN) first generates proposals to tell which regions the grading network head (GNH) should focus upon. The GNH is then used to assign Gleason grades to epithelial cell areas. In the right branch, an Epithelial Network Head (ENH) is used to determine if there is epithelial tissue in the image. The final output depends on the results of the ENH. If there is no epithelial cells, the model outputs the whole image as stroma. Otherwise the model outputs its results from the GNH.

given by

$$L_{obj} = \sum_{i=1}^{N}(-y_i \log(p_i) - (1 - y_i) \log(1 - p_i)) \tag{2.1}$$

where N stands for the total image number in the training datasets; $p_i \in (0, 1)$ is the sigmoid layer output of our model, which can be interpreted as the probability of RoI presence in the image; $y_i \in 0, 1$ is the ground truth of the given image where $y_i = 1$ if the given image has at least one RoI, otherwise $y_i = 0$. Thus, the total loss L of our model is given by

$$L = \underset{ENH}{L_{obj}} + \underbrace{L_{cls} + L_{box} + L_{mask}}_{GNH}. \tag{2.2}$$

2.2.3.3 Transfer Learning

As with most medical image analysis domains, we are limited by a scarcity of accurately annotated training data due to the difficulty and cost of producing high quality data. We compensate for this limitation by using natural image data, which is known as transfer learning. Previous studies have shown that transfer learning in CNNs can alleviate the problem of insufficient training data [12,117]. This is mainly because the learned parameters in the lower layers of neural networks are generic (edges, blobs *etc.*) and can be kept after the pre-training. Thus, transfer learning can help to reduce overfitting on limited medical datasets and allow us to take advantage of networks with more parameters.

Therefore, we utilized an off-the-shelf implementation of Mask R-CNN from Matterport [94], which was trained on the MS COCO dataset [87]. The MS COCO dataset contains more than 200,000 images with pixel-level annotations. Leveraging the effective generalization ability of transfer learning in deep neural networks, we initialized the layers using the pre-trained model followed by fine tuning the ENH and GNH (see details in Section 2.2.3.4).

2.2.3.4 Implementation and Training

Limited by the memory of our GPU, we first cropped our 1200×1200 pixel input image tiles into 16 patches (with overlap) and then downsampled each patch to be 512×512 pixels. These patches, along with their corresponding annotations, were served as the input data for

the training stage. In the testing stage, we again first cropped the images to small patches and then stitched together the network output into the full tiles.

Our main Path R-CNN framework was implemented using the open-source deep learning library Tensorflow [1]. We developed a two-stage training strategy for our model:

- **Stage 1** train the GNH along with the higher layers (stage 4 and 5 in 101 layer structure in [45]) of the ResNet backbone. We used the MS COCO pre-trained model to initialize the network. The network was optimized using stochastic gradient descent (SGD) with backpropagation following the outline of [45]. Adopting a backward fine-tuning strategy, we first trained the GNH for 25 epochs. Then we fine-tuned the ResNet [45] upper layers along with the network head. Figure 2.3 shows a typical training process in Stage 1.

- **Stage 2** takes the fixed weights trained in Stage 1 and only trains the ENH. We chose to fix the Stage 1 weights in this step because of our intuition that epithelial cell detection is a relatively simple task. We empirically found that this method worked very well in practice (see results in Section 2.3.2).

2.2.3.5 Fully Connected Conditional Random Field Post-Processing

After generating predictions from our Path R-CNN model on each image patch, we stitched patches back into the original tiles. This stitching step can lead to artifacting on the edges of each individual patch, as shown in the last two rows of Figure 2.5. We used a fully connected conditional random field (CRF) model to address this problem. This method was first proposed by Krähenbühl *et al.* [67] to compute image segmentations efficiently, which demonstrated the ability to both capture fine edge details and make use of long range dependencies. Chen *et al.* [15] later incorporated this method into CNNs as a post-processing step. A conditional random field $(\boldsymbol{I}, \boldsymbol{X})$ is characterized by $P(\boldsymbol{X}|\boldsymbol{I}) = \frac{1}{Z(\boldsymbol{I})} \exp(-E(\boldsymbol{X}|\boldsymbol{I}))$, where $\boldsymbol{X}$ is defined over the whole image $\{x_1, x_2, \ldots x_N\}$. x_i denotes the label of the i^{th} pixel,

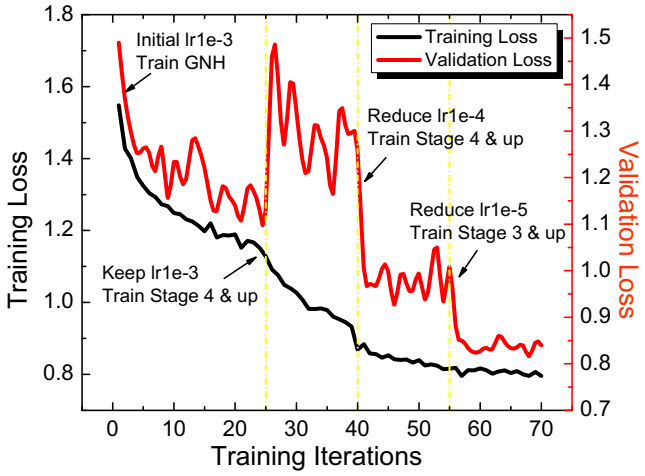

Figure 2.3: The training process to train our proposed model in Stage 1. The model was initialized with the pre-trained weights on MS COCO dataset. The GNH was first trained for 25 epochs with a learning rate of 1e-3. The ResNet stage 4 and upper layers along with GNH were then fine-tuned for 40 epochs with the same learning rate. After convergence of the model parameters, we reduced the learning rate to 1e-4 and trained to 55 epochs. Finally, we included the ResNet stage 3 and fine tuned for another 15 epochs with a learning rate of 1e-5.

N is the total number of pixels. The model employs the energy function

$$E(\boldsymbol{X}|\boldsymbol{I}) = \sum_i \theta_i(x_i) + \sum_{i,j} \theta_{i,j}(x_i, x_j) \tag{2.3}$$

where we refer to first term on the right hand side as the unary potential and the second term as the pairwise potential. The unary potential is defined as $\theta_i(x_i) = -\log P(x_i)$, where $P(x_i)$ is the label assignment probability at pixel i as computed by the segmentation head in the GNH. The pairwise potential is $\theta_{i,j} = \mu(x_i, x_j) \sum_{m=1}^{K} \omega \cdot k^m(\boldsymbol{f_i}, \boldsymbol{f_j})$, where $\mu(x_i, x_j) = 1$ if $x_i \neq x_j$, and zero otherwise. Each k^m is the Gaussian kernel, which depends on features (denoted as $\boldsymbol{f}$) extracted for pixel i and j and is weighted by a learnable parameter ω_m. Following the example of [15], we use bilateral position and color terms in the kernels

$$\omega_1 \exp(-\frac{\|p_i - p_j\|^2}{2\sigma_\alpha^2} - \frac{\|I_i - I_j\|^2}{2\sigma_\beta^2}) + \omega_2 \exp(-\frac{\|p_i - p_j\|^2}{2\sigma_\gamma^2}) \tag{2.4}$$

where p denotes pixel position and I denotes pixel color intensity. Thus, the first kernel term forces nearby pixels with similar color to be in the same class, while the second kernel term removes small isolated regions. The hyperparameters $\sigma_\alpha, \sigma_\beta$ and σ_γ control the "scale" of the Gaussian kernels, which were obtained in the experiment empirically. For simplicity, we refer fully connected CRF as CRF in the later parts of this chapter.

2.2.4 Evaluation Metrics

To make our model comparable with previous work [30,79,111], we use the standard metrics: mean Intersection Over Union (mIOU), Overall Pixel Accuracy (OPA) and Standard Mean Accuracy (SMA) to evaluate the performance of segmentation results. The definition of these metrics is as follows. Assume we have segmentation results f, ground truth label l, and a pixel-wise confusion matrix $\boldsymbol{C}$, where $C_{i,j}$ is the number of pixels labeled as l_i and predicted as f_j. The mIOU is defined as the average of individual Jaccard coefficients, $\mathcal{J}_i$, for all classes l_i. To compute $\mathcal{J}_i$ from the confusion matrix $\boldsymbol{C}$, we use the Jaccard index definition:

$$\mathcal{J}_i = \frac{TP}{TP + FP + FN} = \frac{C_{i,i}}{T_i + P_i - C_{i,i}} \tag{2.5}$$

where $T_i = \sum_{j=1} C_{i,j}$ denotes the total number of pixels with label l_i. $P_j = \sum_i C_{i,j}$ denotes the number of pixels predicted as f_j [18]. The mIOU is then given by

$$\mathcal{J} = \frac{1}{N} \sum^{N} \mathcal{J}_i \tag{2.6}$$

where N is the number of classes. The OPA is defined as

$$OPA = \frac{\sum_i C_{i,i}}{\sum_i \sum_j C_{i,j}}. \tag{2.7}$$

The standard mean accuracy is defined as

$$SMA = \frac{1}{N} \sum_i \frac{C_{ii}}{\sum_j C_{ij}}. \tag{2.8}$$

2.3 Validation Experiments

In this section, we will show our experiment design briefly followed by several experimental results to validate our design for the epithelial cell detection and Gleason grading tasks. The instance segmentation results from the model were converted to semantic segmentation results by choosing the largest probability instance class at each pixel location for the purpose of easy comparison with the previous work.

2.3.1 Experiment Design

We used a ResNet [45] in our Path R-CNN model for feature extraction from the input pathological image. Both the RPN and the GNH adopt a feature pyramid network (FPN) [86] structure by replacing single-scale feature maps with feature pyramids. As in [86], the FPN generates feature pyramids $\{P_2, P_3, P_4, P_5, P_6\}$. For the RPN, we assigned different scale anchors (potential RoIs) $\{32^2, 64^2, 128^2, 256^2, 512^2\}$ at each feature pyramid respectively. The RPN is then trained with the parameters shared across all feature pyramid levels. For the GNH, we assign each RoIs of width w and height h (on the input image to the network) to

the feature pyramid P_k by

$$k = \left\lfloor k_0 + \log_2(\sqrt{wh}/224) \right\rfloor. \tag{2.9}$$

Intuitively, Equation (2.9) means that if the RoI's scale becomes smaller (say, 1/2 of 224), it should be mapped into a finer-resolution level (say, $k = 3$). Through this operation, the model extracts each RoI's information in a similar scale to feed into the GNH. For more implementation detail, we refer readers to [86].

Note that the dataset we have only provides pixel-level annotations. To extract the bounding box of each RoI on the fly, we pick the smallest box that encapsulates all the pixels of the mask. This makes it easy to apply certain image augmentations, such as image rotation, scaling, *etc*., in the pre-processing step.

2.3.2 Results and Discussions

We first discuss quantitative results, which are shown in Table 2.2. We show the averaged performance (measured by OPA, SMA and mIOU) of our proposed method as well as of different baseline methods on our dataset. We then show the results of ablation studies that analyze the effect of adding the ENH and CRF to our framework.

2.3.2.1 5-fold Cross Validation

For our tile-based model evaluation, the full 513 image tileset was randomly divided into 5 non-overlapping cross validation folds. During training, we observed quick convergence when using pre-trained weights trained on MS COCO dataset. Table 2.2 (Row 3) and Figure 2.4 show the performance of our model. Our model achieves 79.56% mIOU, 88.78%SMA, and 89.40% OPA among the four classes. In these four classes, Path R-CNN has a relatively good performance in "stroma", "benign", and "high-grade" classification. However, it only achieves 79.54% IOU for "low-grade". This is because of the large appearance variance of "low-grade" glands. In "low-grade", the glands differ in size and shape, and are often long

Table 2.2: Model performance on segmenting prostate histological images as "Stroma" (BG), "Benign" (BN), "Low-Grade" (LG), and "High-Grade" (HG).

	J_{BG}	J_{BN}	J_{LG}	J_{HG}	$mIOU$	OPA	SMA
Handcrafted [30]	59.5%	35.2%	49.5%[1]	N/A	48.1%	N/A	N/A
Multi-Scale U-Net [79]	82.42%	72.13%	58.70%	78.38%	72.91%	87.30%	86.04%
FCN-8s [52]	N/A	N/A	N/A	N/A	75.9%	87.3%	N/A
Path R-CNN	**83.14%**	**83.87%**	**71.54%**	**79.69%**	**79.56%**	**89.40%**	**88.78%**
Path R-CNN w/o ENH	73.26%	75.71%	71.13%	71.57%	72.91%	84.13%	86.19%
Path R-CNN w/o CRF	82.94%	83.63%	71.32%	79.48%	79.34%	89.26%	88.70%

and/or angular. They are usually micro-glandular, however, some may be medium to large in size. This size and shape variation can be easily seen in the second column of Figure 2.4, where "low-grade" glands are shown by the green color.

2.3.2.2 Model Comparison

We compared our model with several baseline models. For the standard and multi-scale U-Net models, pixel-wise confusion matrices were summed across all 5 folds. Results from a support vector machine and random forest model based on handcrafted features [30] are also reported in Table 2.2. Note that the IOU of the random forest model for "Low-Grade" class is calculated by combining "Low-Grade" and "High-Grade" together, as done in their paper. Our proposed Path R-CNN achieved the highest performance in both the single class evaluation and the four class mIOU. We credit the performance improvement to the following five differences between our model and the baseline models. First, we adopted a two-stage approach in the left branch. Using the recently popular concept of neural networks with "attention" mechanisms, the RPN module (1^{st} stage) tells the GNH module (2^{nd} stage) where to focus. Second, compared to previous efforts that used a simple segmentation mask as the ground truth label, we extracted and provided more information (cancer ROI location,

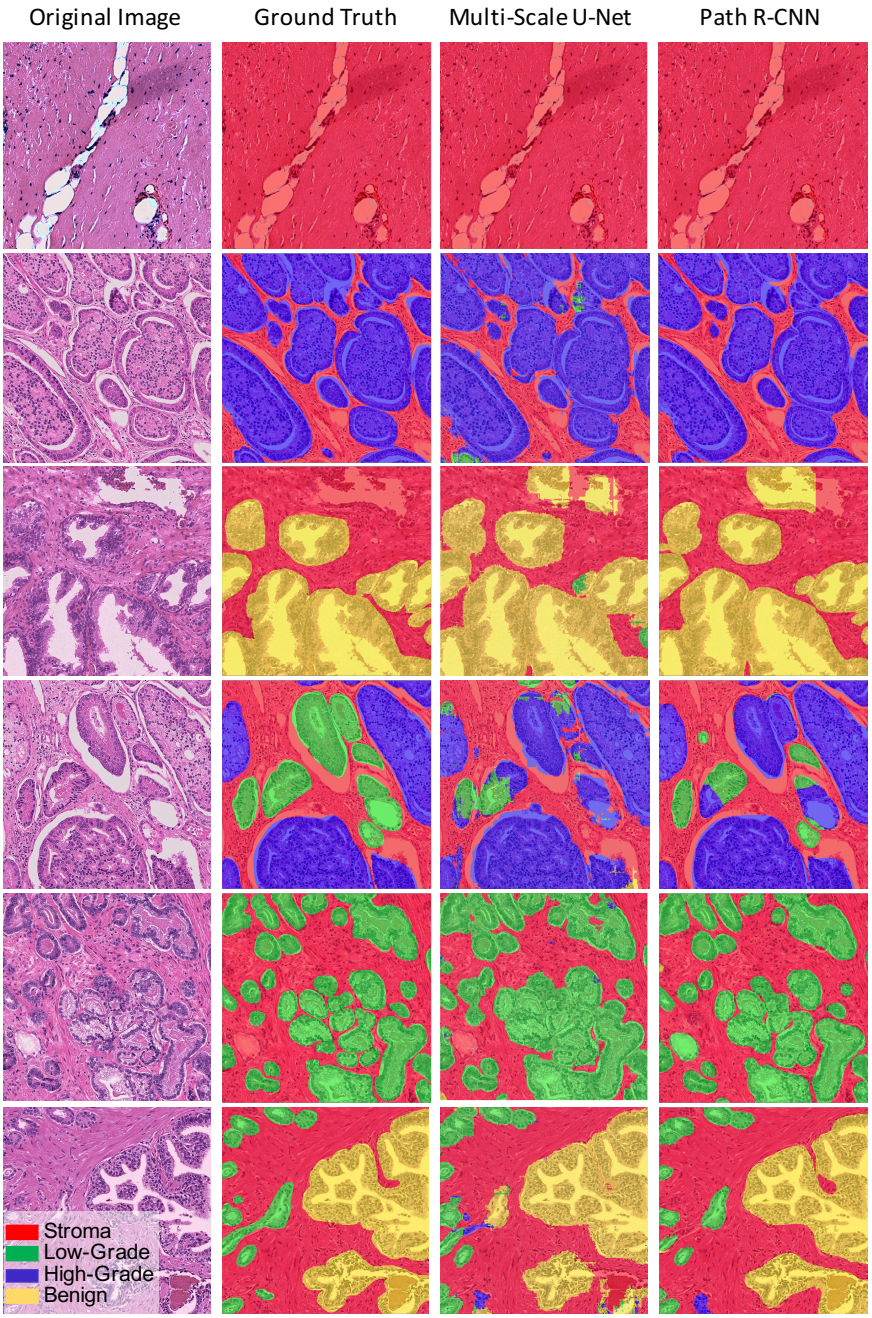

Figure 2.4: Path R-CNN model results. **(Left Column):** Original histological image tiles stained by H&E. **(Middle Left Column):** Slides annotated by pathologist experts served as the ground truth to train Path R-CNN. **(Middle Right Column)**: Multi-Scale U-Net Predictions. **(Right Column):** Path R-CNN Predictions.

shape, and aggressiveness) to the network by using a multi-task framework. Training different tasks simultaneously using the GNH module helped regularize the network. Third, by adding the ENH to the framework, we solved the issue of models commonly predicting cancer areas in images consisting entirely of stroma, which helped boost performance by a large margin. Fourth, we used a large neural network, ResNet, for image feature extraction. ResNet was able to take advantage of a large number of parameters while avoiding the degradation problem [45]. Fifth, the GNH decouples the segmentation task and classification task, which proved to be key in boosting model performance [44].

2.3.2.3 ENH Effect

Here, we analyze the important role that the ENH played in our system.

We first formulated our network as a multi-task framework that minimizes a multi-task loss function (Equation 2.2) simultaneously. However, this formulation did not yield substantial improvement over the baseline model [79]. We hypothesize two possible reasons for this: 1) The objectness prediction loss shown in Equation (2.1) for ENH, which is a per-image loss, is not within the same scale as the other losses, and 2) The ENH might interfere with the GNH in a complex manner that lowers the performance of every task when trained simultaneously. To solve this problem, we adopt a two-stage training approach as stated in Section 2.2.3.4 under the assumption that epithelial cell detection is a relatively simple task.

To measure the performance of the ENH, we calculated the area under the curve (AUC) of the receiver operating characteristic (ROC) curve using the same 5-fold cross-validation method described previously. The ENH had superb performance, with an AUC of $0.9984 \pm$ 1.329e-3. This result demonstrates that epithelial cell detection can be performed robustly using the simple network structure of the ENH.

We also demonstrate the mIOU results without the ENH in Row 5 of Table 2.2 and the first two rows of Figure 2.5. By comparing the results of Row 4 and Row 5 in Table

[1]The previous model by Gertych, *et al*. [30] only addressed three class segmentation by combining G3 and G4 together.

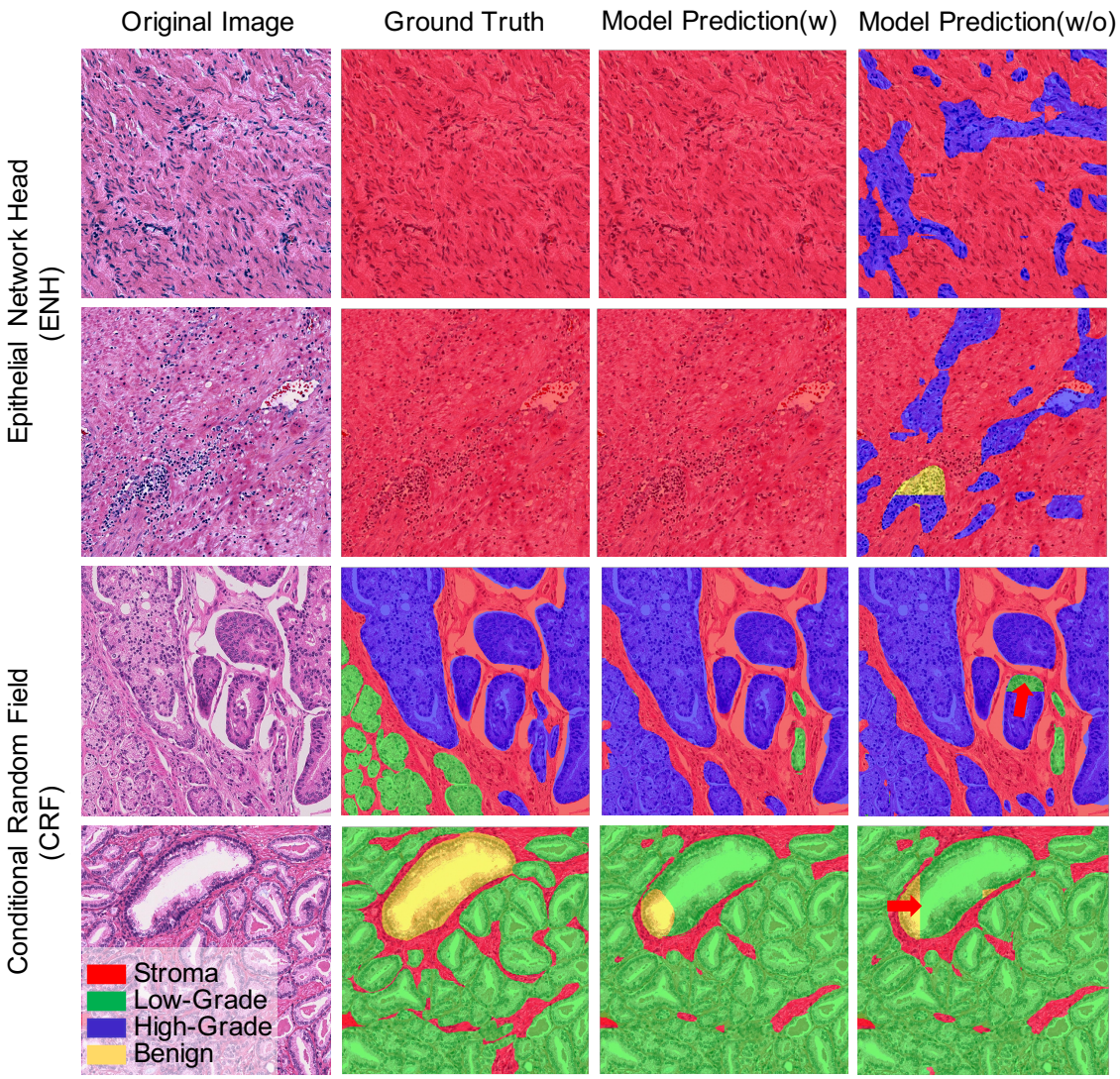

Figure 2.5: Effectiveness of adding the ENH and CRF to our proposed Path R-CNN. The first two rows show two examples to demonstrate the effectiveness of the ENH. The last two rows show two additional examples to demonstrate the effectiveness of adding the CRF.

2.2, we see that the ENH boosts the segmentation performance by a large margin. This is mainly because of the trade-off between objectness prediction accuracy and the segmentation accuracy in our model settings. Without ENH, if we want our system to have a high precision that minimizes failure to detect potential epithelial areas, we need to lower the detection threshold. This will give us a model that is intended to predict epithelial cells more often even in an image that is full of stroma; thus the performance will be reduced dramatically. This can been observed in the first two rows of Figure 2.5. In the last column, we see that the model is prone to predict ROIs in large areas of stroma. Thus, we conclude that the ENH is crucial for achieving good performance in our system. Additional rationale and advantages of the ENH are discussed in the Supplementary Information.

2.3.2.4 Post-Processing using CRF

Our results using the CRF show that adding the method helps remove unnatural boundaries created by stitching, as shown in last two rows of Figure 2.5. The red arrows in the figure (Row 3 and 4) indicate the unnatural boundaries output by the stitching process. After CRF post-processing, we observe these unnatural boundaries are removed. The CRF also helps improve mIOU slightly, as shown in Row 6 of Table 2.2.

2.4 Limitations and Future Work

Here, we discuss some limitations of our work and provide potential research directions that could help address these limitations.

We note that the 5-fold validation used in our experiments is not a patient-wise validation. Unfortunately, we did not have patient-level information with which to perform a more rigorous patient-level stratification. This might result in a positive bias since a cancer can look similar in tiles within the same patient, especially in tiles that are spatially close to one another. However, we argue that relative model comparisons in this work are fair as we used the exactly same train-test data split as in [52] across all models.

Additional careful tuning of the loss scale of $L_{cls}, L_{box}, L_{mask}, L_{obj}$ could allow all training to happen simultaneously (rather than in two stages) by achieving a better balance of trade-offs between the losses. In this case, a single end-to-end training process could be achieved for the system.

Another area for potential improvement is the "RoIAlign" layer. The "RoIAlign" layer [44] extracts a small feature map from the corresponding feature pyramid layer for each RoI right before the network head by using Equation 2.9. It results in the loss of some scale information which might be important for histopathology. In particular, this information might be helpful for the Gleason grading task as different sizes of glands can be categorized into levels in the Gleason system. Therefore, incorporating scale information in the GNH might be helpful to improve the system's performance.

Finally, we re-examined those individual images upon which our system performed worst. We found in some of these images that there were intrinsic difficulties that even expert pathologists might not agree upon. If we were to treat our model as another pathologist, some experts might agree with its predictions while others might not. This observation leads to bigger questions: how do we best form a "Doctor-AI Ecosystem"? How might the experts' annotations affect the training of computer systems? How do our computer systems' performance affect doctors' decisions in practice? And what is a good criterion that we can use to tell if computer systems are trustworthy enough to make their diagnosis alone [129]. Those are the questions we need to answer in the future.

2.5 Conclusions

In this chapter, we present a novel framework that achieved state-of-the-art performance in epithelial cell detection and Gleason grading based on histological images. We adopted a two-stage model, R-CNN, to help the network focus on regions that need a careful inspection. By adding an Epithelial Network Head (EHN), our model performance was boosted by detecting epithelial cells and predicting Gleason grades simultaneously with little additional overhead.

We also employed a fully connected conditional random field (CRF) as a post-processing step to compensate for the artifacts caused by the system. Extensive experiments were conducted to validate the robustness of our method and the effectiveness of each module in our model. We envision that our method would help the pathologist to make the diagnosis more efficiently in the near future.

CHAPTER 3

Initial Exploration on Semi-supervised Learning Using Generative Adversarial Network

3.1 Introduction

In this chapter, we start to move our focus to semi-supervised learning (SSL). In particular, we investigate semi-supervised learning methods based on generative adversarial networks (GANs), which have received much attention. Among them, two distinct approaches have achieved competitive results on a variety of benchmark datasets. Bad GAN learns a classifier with unrealistic samples distributed on the complement of the support of the input data. Conversely, Triple GAN consists of a three-player game that tries to leverage good generated samples to boost classification results. In this chapter, we perform a comprehensive comparison of these two approaches on different benchmark datasets. We demonstrate their different properties on image generation, and sensitivity to the amount of labeled data provided. By comprehensively comparing these two methods, we hope to shed light for our GAN-based semi-supervised learning in the next chapter.

3.1.1 Motivation

Semi-supervised learning (SSL) aims to make use of large amounts of unlabeled data to boost model performance, typically when obtaining labeled data is expensive and time-consuming. Various semi-supervised learning methods have been proposed using deep learning and proven to be successful on several standard benchmarks. Weston *et al.* [138] employed a manifold embedding technique using the pre-constructed graph of unlabeled data;

Rasmus *et al*. [107] used a specially designed auto-encoder to extract essential features for classification; Kingma and Welling [63] developed a variational auto encoder in the context of semi-supervised learning by maximizing the variational lower bound of both labeled and unlabeled data; Miyato *et al*. [97] proposed virtual adversarial training (VAT) that tied to find a deep classifier, which had a good prediction accuracy on training data and meanwhile was less sensitive to data perturbation towards the adversarial direction.

In recent years, generative adversarial networks (GANs) [37], have demonstrated their capability in SSL frameworks [16, 19, 29, 71, 75, 83, 113]. GANs are a powerful class of deep generative models that are able to model data distributions over natural images [95, 106]. Salimans *et al*. first proposed to use GANs to solve a $(K+1)$-class classification problem, where the dataset contained K class originally and the additional $(K+1)$th class consisted of the synthetic images generated by the GAN's generator. Later on, Li *et al*. [16] realized that the generator and discriminator in [113] may not be optimal at the same time (*i.e*., the discriminator was able to achieve good performance in SSL, while the generator may generate visually unrealistic images). They proposed a three-player game (Triple-GAN) to simultaneously achieve good classification results and obtained a good image generator. Dai *et al*. [19] realized the same problem, but instead gave theoretical justifications of why using bad samples from the generator was able to boost SSL performance. Their model is called Bad GAN, which achieves state-of-the-art performance on multiple benchmark datasets. Another line of work focused on manifold regularization [7]. Kumar *et al*. [71] estimated the manifold gradients at input data points and added an additional regularization term to a GAN, which promoted invariance of the discriminator to all directions in the data space. Lecouat *et al*. [75] performed manifold regularization by approximating the Laplacian norm that was easily computed within a GAN and achieved competitive results.

In this chapter, we focus on two GAN-based SSL models, Triple GAN and Bad GAN, and perform a comprehensive comparison between them. For simplicity, we refer to Triple GAN as Good GAN in contrast to Bad GAN.

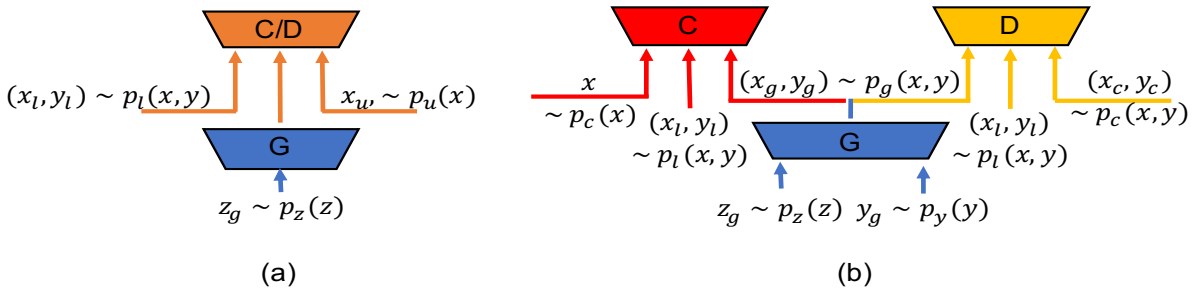

Figure 3.1: Network architecture of Bad GAN (a) and Good GAN (b). Bad GAN (a) consists of two parts: a generator G aims to generates "bad" samples, and a discriminator/classifier D/C that distinguishes real and fake samples and put the labeled samples into the right classes; Good GAN (b) consists of three parts: two conditional networks G and C that generate pseudo labels given real data and pseudo data given real labels respectively, and a separate discriminator D that distinguish the generated data-label pair from the real data-label pair.

3.1.2 Related Works

3.1.2.1 Bad GAN

Suppose we have a classification problem that requires classifying a data point $\boldsymbol{x}$ into one of K possible classes. A standard classifier takes in $\boldsymbol{x}$ as input and outputs a K-dimensional vector of logits $\{l_1, ..., l_K\}$. Salimens *et al.* [113] extend the standard classifier C by simply adding samples from the GAN generator G to the dataset, labeling them as a new "generated" class $y = K + 1$, and correspondingly increasing the dimension of C output from K to $K + 1$. The loss function $L_{C/D}$ for training C (*i.e.*, the extended discriminator D from the GAN's

perspective) then becomes

$$
\begin{aligned}
L_{C/D} &= L_{\text{supervised}} + L_{\text{unsupervised}} \\
L_{\text{supervised}} &= \mathop{\mathbb{E}}_{\boldsymbol{x},y\sim p_l(\boldsymbol{x},y)}[-\log(p_{C/D}(y|\boldsymbol{x}, y < K+1))] \\
L_{\text{unsupervised}} &= \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p_u(\boldsymbol{x})}[-\log(1 - p_{C/D}(y = K+1|\boldsymbol{x}))] \\
&\quad + \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p_g(\boldsymbol{x})}[-\log(p_{C/D}(y = K+1|\boldsymbol{x}))]
\end{aligned}
\tag{3.1}
$$

The supervised loss term $L_{\text{supervised}}$ is a traditional cross-entropy loss that is applied to labeled data $(\boldsymbol{x}, y) \sim p_l(\boldsymbol{x}, y)$. The unsupervised loss requires C/D to put the synthetic data from generator $\boldsymbol{x} \sim p_g(\boldsymbol{x})$ into the $(K+1)$th class, while putting the unlabeled data $\boldsymbol{x} \sim p_u(\boldsymbol{x})$ into the real K classes. For the generator, [113] found feature matching loss in Eq. 3.2 is the best in practice, though they generated visually unrealistic images. The feature matching loss is,

$$
L_G = \left\| \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p_u}(\boldsymbol{f}(\boldsymbol{x})) - \mathop{\mathbb{E}}_{\boldsymbol{z}_g\sim p_z(z)}(\boldsymbol{f}(G(\boldsymbol{z}_g))) \right\|_2^2 \tag{3.2}
$$

where $\boldsymbol{z}_g \sim p_z(\boldsymbol{z})$ is drawn from a simple distribution such as uniform.

On the basis of this formulation, Dai *et al.* [19] give a theoretical justification on why the visually unrealistic images (*i.e.*, "bad" samples) from the generator could help with SSL. Loosely speaking, the carefully generated "bad" samples along with the loss function design in Eq. 3.1 could force C's decision boundary to lie between the data manifolds of different classes, which in turn improves generalization of the classifier. Based on this analysis, they propose a Bad GAN model that learns a bad generator by explicitly adding a penalty term to generate "bad" samples. Their objective function of the generator becomes:

$$
\begin{aligned}
L_G = &-\mathcal{H}[p_g(\boldsymbol{x})] + \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p_g(\boldsymbol{x})}(\log p^{pt}(x)\, \mathbb{I}[p^{pt}(x) > \epsilon] \\
&+ \left\| \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p_u(\boldsymbol{x})}(\boldsymbol{f}(\boldsymbol{x})) - \mathop{\mathbb{E}}_{\boldsymbol{z}_g\sim p_z(z)}(\boldsymbol{f}(G(\boldsymbol{z}_g))) \right\|_2^2
\end{aligned}
\tag{3.3}
$$

where the first term measures the negative entropy of the generated samples and tries to avoid collapsing while increasing the coverage of the generator. The second term explicitly penalizes generated samples that are in high density areas by using a pre-trained model, and the third term is the same feature matching term as in Eq. 3.2.

3.1.2.2 Good GAN

Li *et al*. [16] also noticed the same problem in [113] as the generator and the discriminator have incompatible loss functions, but took a different approach to tackling this issue. Intuitively, assume the generator can generate good samples in the original settings of [113], the discriminator should identify these samples as fake samples as well as predict the correct class for the generated samples. To address the problem, [16] present a three-player game called Triple-GAN that consists of a generator G, a discriminator D, and a separate classifier C. C and D are two conditional networks that generate pseudo labels given real data and pseudo data given real labels respectively. To jointly evaluate the quality of the samples from the two conditional networks, a single discriminator D is used to distinguish whether a data–label pair is from the real labeled dataset or not. We refer this model as Good GAN because one of the aims for this formulation is to obtain a good generator.

The authors prove that instead of competing equilibrium states as in [113], Good GAN has the unique global optimum for both C and G, *i.e.*, $p(\boldsymbol{x}, y) = p_g(\boldsymbol{x}, y) = p_c(\boldsymbol{x}, y)$, the three joint distributions match one another. In other words, a good classifier will result in a good generator and vice versa. Furthermore, Good GAN is trained using the REINFORCE algorithm, in which it generates pseudo labels through C for some unlabeled data and uses these pairs as positive samples to feed into D. This is a key to the success of the model, as one of the crucial problems of SSL is the limited size of the labeled data. Figure 3.1 shows the network architecture of Good GAN and Bad GAN.

3.1.3 Contributions

In this chapter, we systematically and extensively compared two GAN-based SSL methods, Good GAN and Bad GAN, by applying these two models with commonly-used benchmark datasets. As both of models attempt to solve a similar issue in the original setting [113] but are motivated by dissimilar perspectives, we believe that our comparison will provide insight for future SSL research, including our proposed UGAN model in the next chapter.

3.1.4 Organizations

The rest of the chapter is organized as follows. In Section 3.2, we show the network architecture we employed, benchmark datasets we used, and hyperparameters we selected in order to perform a fair comparison between these two models; in Section 3.3, we demonstrate our comparison results and discuss several important aspects we found for these two models; we conclude this chapter in Section 3.4.

3.2 Methods

3.2.1 Network Architecture

To perform a fair comparison between Good GAN and Bad GAN, we use the same network architecture for the generator G and the classifier C in both models. We follow the architecture closely in [16] to set up the additional discriminator D in Good GAN. Both of them use Leaky-Relu activation and weight normalization to ease the difficulty of GAN's training. Implementing them using same architecture ideally avoids the possibility of using an architecture that is custom-tailored to work well with one or the other. Detailed model architectures can be found in the Appendix B.1.

3.2.2 Datasets

Using the above-defined network architectures, we compare the two models on the widely adopted MNIST [77], SVHN [99], and CIFAR10 [69] datasets. MNIST consists of 50,000 training samples, 10,000 validation samples, and 10,000 testing samples of handwritten digits of size 28×28. SVHN consists of 73,257 training samples and 26,032 testing samples. Each sample is a colored image of size 32×32, containing a sequence of digits with various backgrounds. CIFAR10 consists of colored images distributed across 10 general classes – *airplane*, *automobile*, *bird*, *cat*, *deer*, *dog*, *frog*, *horse*, *ship* and *truck*. It contains 50,000 training samples and 10,000 testing samples of size 32×32. Following [16], we reserve 5,000 training samples from SVHN and CIFAR10 for validation if needed. For our CIFAR10 experiment, we perform zero-based component analysis (ZCA) [73] as suggested in [16] for the input of C, but still generate and estimate the raw images using G and D.

We perform an extensive investigation by varying the amount of labeled data. Following common practice, this is done by throwing away different amounts of the underlying labeled dataset [104,112,113,132]. The labeled data used for training are randomly selected stratified samples unless otherwise specified. We perform our experiments on setups with 20, 50, 100, and 200 labeled examples in MNIST, 500, 1000, and 2000 labeled examples in SVHN, and 1000, 2000, 400, 8000 examples in CIFAR10.

3.2.3 Hyperparameter Selection

For the hyperparameter selection such as learning rate and beta for Adam optimization, and the coefficient for each cost function term, we closely follow [16, 19]. In addition, we perform extensive study of the effects of batch size on performance for Bad GAN. As reported by [75], Bad GAN training is sensitive to training batch size, and thus we vary batch size in the training phase and compare their final performances on MNIST and SVHN.

Table 3.1: Test accuracy on semi-supervised MNIST. Results are averaged over 10 runs. * denotes the special selection of labeled data. See details in Section 3.3.3.

Model	Test accuracy for a given number of labeled samples			
	20	50	100	200
Bad GAN [19]	-	-	**99.21 ± 0.01%**	-
Triple GAN [16]	95.19 ± 4.95%	98.44 ± 0.72%	99.09 ± 0.58%	**99.33 ± 0.16%**
Bad GAN (ours)	68.12 ± 0.60%	96.24 ± 0.16%	99.17 ± 0.03%	99.20 ± 0.03%
Good GAN (ours)	**95.93 ± 4.45%***	**98.68 ± 1.12%**	99.07 ± 0.46%	99.17 ± 0.08%

Table 3.2: Test accuracy on semi-supervised SVHN. Results are averaged over 10 runs.

Model	Test accuracy for a given number of labeled samples		
	500	1000	2000
Bad GAN [19]	-	**95.75 ± 0.03%**	-
Triple GAN [16]	-	94.23 ± 0.17%	-
Bad GAN (ours)	94.21 ± 0.45%	95.32 ± 0.07%	**95.47 ± 0.39%**
Good GAN (ours)	**94.67 ± 0.12%**	95.30 ± 0.38%	95.37 ± 0.09%

Table 3.3: Test accuracy on semi-supervised CIFAR10. Results are averaged over 10 runs.

Model	Test accuracy for a given number of labeled samples			
	1000	2000	4000	8000
Bad GAN [19]	-	-	**85.59 ± 0.03%**	-
Triple GAN [16]	-	-	83.01 ± 0.36%	-
Bad GAN (ours)	77.58 ± 0.17%	81.36 ± 0.08%	82.89 ± 0.13%	**85.47 ± 0.10%**
Good GAN (ours)	**81.08 ± 0.57%**	**81.79 ± 0.37%**	82.82 ± 0.41%	85.37 ± 0.18%

3.3 Experimental Results and Discussion

We implement Good GAN based on Tensorflow 1.10 [31] and Bad GAN based on Pytorch 1.0 [103]. The generated images from gG is not applied until the number of epochs reach a threshold that gG could generate reliable image-lable pairs. We choose 200 in all three cases. All of the other hyperparameters including initial learning rate, maximum epoch number, relative weights and parameters in Adam [62] are fixed according to [16, 19, 113] across all of the experiments.

3.3.1 Classification

We report our classification accuracy on the test set in Table 3.1, Table 3.2 and Table 3.3 for MNIST, SVNH and CIFAR10, respectively, along with the results reported in the original papers. The similarity of our results to those reported in the original papers suggests that our reproduced models are accurate instantiations of Good GAN and Bad GAN. Furthermore, we perform extensive study by varying the amount of labeled data and observe that Good GAN and Bad GAN behave quite differently under various circumstances.

First, with a medium amount of labeled data (*e.g.*, MNIST with 100 or 200 labeled data, SVHN with more than 2000 labeled data, or CIFAR10 with more than 2000 labeled data), Bad GAN performs better than Good GAN. In fact, to the best of our knowledge, Bad GAN achieves the current state-of-the-art performance on those benchmark datasets. However, with low amounts of labeled data, Good GAN performs better, which demonstrates that Good GAN is less sensitive to the amount of labeled data than Bad GAN. One possible explanation is due to the use of the REINFORCE algorithm in Good GAN, because it generates pseudo labels through C for some unlabeled data and uses these pairs as positive samples of D. Since C converges quickly, this trick provides a clever way to enable the generator to explore a much larger data manifold that includes both the labeled and unlabeled data information. In other words, the classifier is able to provide pseudo labels for the unlabeled data, while the discriminator will judge if the pseudo labels are reliable or not

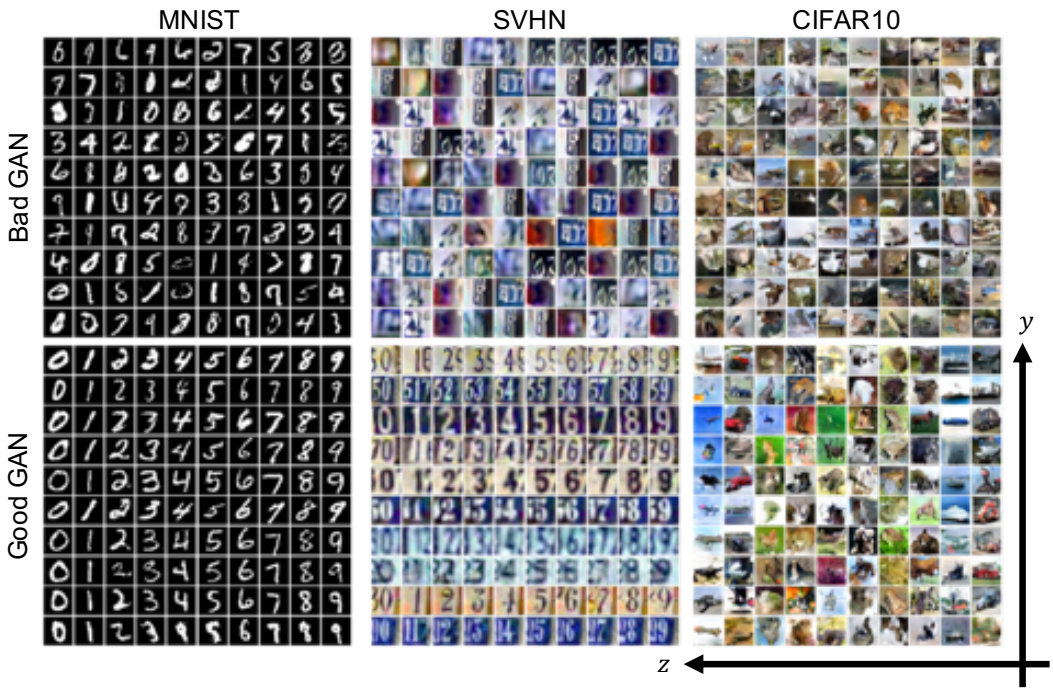

Figure 3.2: Generated images from both Bad GAN (top) and Good GAN (bottom). The images generated from Good GAN are produced by varying the class label y in the vertical axis and the latent vector z in the horizontal axis.

throughout the training. This in return will affect the evolution of the generator, which will take advantage of the unlabeled data to generate good images. Generated good image-label pairs that implicitly contain unlabeled data information will eventually benefit the classifier. This works extremely well for relatively simple datasets like MNIST, as Good GAN is able to model the class-awarded data distribution through weak supervision. On the other hand, Bad GAN yields decreased performance when the amount of labeled data is low, as it does not have any mechanism to augment the information that could be used to train the classifier in this case.

3.3.2 Generated Images

In Figure 3.2, we compare the quality of images generated by Good GAN and Bad GAN. As can be seen, Good GAN is able to generate clear images and meaningful samples conditioned

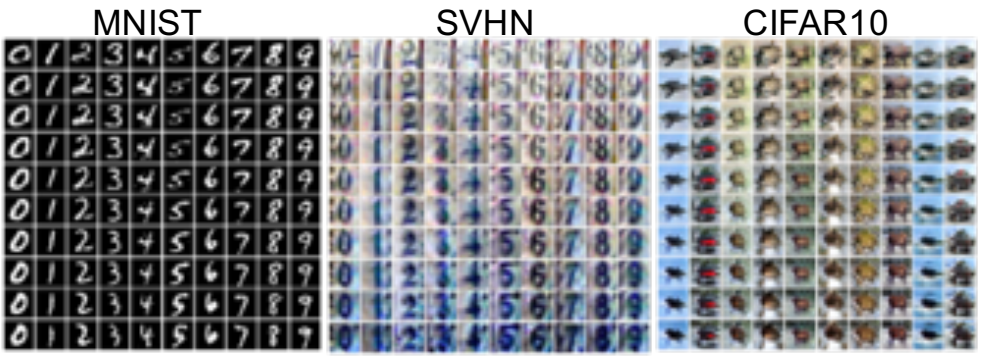

Figure 3.3: Class-conditional latent space interpolation. We first sample two random latent vectors z and linearly interpolate them. Then we map these vectors to the image space conditioned on each class y. The vertical axis is the direction for latent vector interpolation while the horizontal axis is the direction for varying the class labels.

on class labels, while Bad GAN generates "bad" images that look like a fusion of samples from different classes. In addition, Good GAN is able to disentangle classes and styles. In Figure 3.2 bottom, we vary the class label y in the vertical axis and the latent vectors z in the horizontal axis to generate the images. As shown in the figure, the latent vector z encodes meaningful physical appearances, such as scale, intensity, orientation, color and so on, while the label y controls the semantics of the generated images. Furthermore, Good-GAN can transition smoothly from one style to another with different visual factors without losing the label information as shown in Figure 3.3. This proves that Good GAN can learn meaningful latent space representations instead of simply memorizing the training data.

3.3.3 Importance of Selection of Labeled Data

Another interesting observation is that the selection of labeled data plays a crucial role for training the Good GAN model in the low labeled data scenario. As mentioned above, the labeled data used for the training are randomly selected stratified samples, except for the MNIST-20 case. In this case, we found selecting representative labeled data to train is the key to achieving good performance. The reported accuracy in Table 3.1 is averaged over

Figure 3.4: Two-runs of Good GAN model on MNIST dataset. (a) A single run where we randomly select 20 labeled data. The generator generates a lot of wrong images conditioned on the label and the classifier has lower performance. (b) Another run where we manually select 20 representative labeled examples. This time the generator is able to generate correct images, and the classifier achieves good classification performance.

10 runs where we manually selected different representative labeled data in a stratified way. Figure 3.4 (a) shows a single run that uses randomly selected labeled data and does not achieve good results, while Figure 3.4 (b) shows another run that is able to achieve higher accuracy. The failure of the first run is due to the initial selections for digit 4 being similar to 9, causing the generator to generate many 9s when conditioned on label 4. The generator also generates low-quality images. We also report that with a random selection of 20 labeled data, the Good GAN was able to achieve $76.78 \pm 6.47\%$ accuracy over 3 runs.

3.3.4 Importance of Batch Size

We found that batch size significantly affects the final training results, in both Good GAN and Bad GAN. To investigate the effect of batch size on Bad GAN performance, we performed experiments with different batch sizes on MNIST (with 100 labeled samples) and SVHN (with 1000 labeled samples) using Bad GAN. As shown in Table 3.4, we empirically show that the performance of Bad GAN is sensitive to training batch size, and the optimal performance for each dataset is achieved with a batch size of 100.

To further understand the effect of the batch size on Bad GAN training, we present the generator loss with different batch sizes for MNIST and SVHN in Figure 3.5. The results indicate that smaller batch sizes lead to larger generator loss in the final stage of training. As that generator loss mainly depends on the first-order feature matching loss in Bad GAN, an intuitive explanation could be that larger batch sizes reduce the variance of the sample mean, allowing the generator to quickly approximate the entire training set. This leads to smaller generator loss, especially when model training becomes more stable in the final stage.

As noted by [19], feature matching is performing distribution matching in a weak manner, which could be significantly affected by batch size. On one extreme, when the batch size is too small, the power of the generator in distribution matching is weak due to the excessive generator loss. Generated samples are therefore more likely to diverge from the manifold. Especially when data complexity increases, it is more difficult to minimize the KL divergence between the generator distribution and a desired complement distribution in Bad GAN, which could be one possible reason why model degradation is more significant on SVHN when using 20 batch size. On the other extreme, larger batch size leads to smaller generator loss, which comes with reduced diversity of generated samples. When the batch size is too large, the small generator loss will lead to a collapsed generator which fails to generate diverse samples that cover complement manifolds. As a result, the decision boundary between such missing manifolds becomes under-determined, which will also degrade model performance. We plot Bad GAN performance under different batch sizes for MNIST and SVHN in Appendix B.2.

Based on our experience, Good GAN is best when we use a large batch size. Intuitively, a small batch size is not good for the REINFORCE algorithm adopted in Good GAN because a single wrong prediction of the unlabeled data will have a big impact on the weight update in each iteration. We perform Good GAN experiments on SVHN using different batch sizes. The results are shown in Table 3.5. Empirically, we find that with small batch size, Good GAN is not able to generate good image-label pairs, hence the generated image-label pairs even hurt the classifier's performance when we use them to train. (See in Appendix B.2).

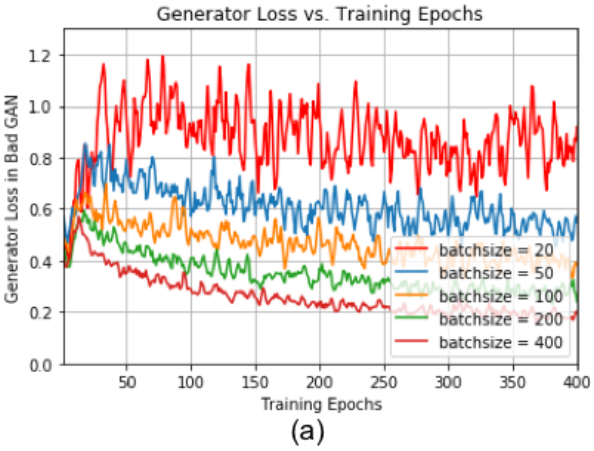

Figure 3.5: Batch size effect on generator loss in Bad GAN. The experiments are performed on (a) MNIST using 100 labeled samples and (b) SVHN using 1000 labeled samples.

Table 3.4: Bad GAN performance versus batch size on MNIST and SVHN. The results are achieved using 100 labeled samples in MNIST and 1000 labeled samples in SVHN.

Batch size	20	50	100	200	400
MNIST-100	$98.90 \pm 0.04\%$	$99.10 \pm 0.03\%$	$\mathbf{99.17 \pm 0.03}\%$	$99.16 \pm 0.03\%$	$98.89 \pm 0.02\%$
SVHN-1000	$93.35 \pm 0.05\%$	$95.29 \pm 0.03\%$	$\mathbf{95.56 \pm 0.02}\%$	$95.19 \pm 0.02\%$	$94.20 \pm 0.04\%$

Table 3.5: Good GAN performance versus batch size on SVHN. The results are achieved using 1000 labeled samples in SVHN.

Batch size	20	50	100
SVHN-1000	92.47%	92.59%	**95.30**%

3.4 Conclusions

In this chapter, we systematically and extensively compared two GAN-based SSL methods, Good GAN and Bad GAN, by applying these two models with commonly-used benchmark datasets. We illustrate the distinct characteristics of the images they generated, as well as each model's sensitivity to varying the amount of labeled data used for training. In the case of low amounts of labeled data, model performance is contingent on the selection of labeled samples; that is, selecting non-representative samples results in generating incorrect image-label pairs and deteriorating classification performance. Furthermore, selecting the optimal batch size is crucial to achieve good results in both models. Notably, Good GAN and Bad GAN models can be used for complementary purposes; Good GAN generates good image-label pairs to train the classifier, while Bad GAN generates samples that force the decision boundary between data manifold of different classes. We envision that combining these two methods should yield further performance improvement in SSL.

CHAPTER 4

Semi-supervised Learning using Adversarial Training with Good and Bad Samples

4.1 Introduction

In this chapter, we investigate semi-supervised learning (SSL) for image classification using adversarial training. Previous results have illustrated that generative adversarial networks (GANs) can be used for multiple purposes in SSL . Triple-GAN, which aims to jointly optimize model components by incorporating three players, generates suitable image-label pairs to compensate for the lack of labeled data in SSL with improved benchmark performance. Conversely, Bad (or complementary) GAN, optimizes generation to produce complementary data-label pairs and force a classifier's decision boundary to lie between data manifolds. Although it generally outperforms Triple-GAN, Bad GAN is highly sensitive to the amount of labeled data used for training. Unifying these two approaches, we present unified-GAN (UGAN), a novel framework that enables a classifier to simultaneously learn from both good and bad samples through adversarial training. We perform extensive experiments on various datasets and demonstrate that UGAN: 1) achieves competitive performance among other GAN-based models, and 2) is robust to variations in the amount of labeled data used for training.

4.1.1 Motivation

With recent progress in deep learning, large labeled training datasets are becoming increasingly important [2, 20, 68, 87]. However, labeling such datasets is expensive and time-

consuming. Semi-supervised learning (SSL) aims to leverage large amounts of unlabeled data to boost model performance. Various SSL methods have been proposed using deep learning and proven to be successful. Weston *et al*. [138] employed a manifold embedding technique using a pre-constructed graph of unlabeled data; Rasmus *et al*. [107] used a specially designed auto-encoder to extract essential features for classification; Kingma and Welling [63] developed a variational auto encoder by maximizing the variational lower bound of both labeled and unlabeled data; Miyato *et al*. [97] proposed virtual adversarial training (VAT), which helped find a deep classifier that had a good prediction accuracy and was less sensitive to data perturbation towards the adversarial direction.

Recently, generative adversarial networks (GANs) [37], have demonstrated their capability in SSL frameworks [16,19,29,71,75,83,113]. GANs are a powerful class of deep generative models that can represent data distributions over natural images [95, 106]. Specifically, a GAN is formulated as a two-player game, where the generator G takes a random vector z as input and produces a sample $G(z)$ in the data space, while the discriminator D identifies whether a certain sample comes from the true data distribution $p(x)$ or the generator. As an extension, Salimans *et al*. [113] first proposed feature-matching GANs (FM-GANs) to solve an SSL problem. Suppose we have a classification problem that requires classifying a data point x into one of K possible classes. A standard classifier takes x as input and outputs a K-dimensional vector of logits $\{l_1, ..., l_K\}$. Salimans *et al*. extended the standard classifier by simply adding samples from a GAN's G to the dataset, labeling them as a new "generated" class $y = K + 1$, and correspondingly increasing the classifier's output dimension from K to $K + 1$. They also found that using feature matching loss in G improved classification performance. The $(K+1)$-class discrimination objective with feature matching loss in G led to strong empirical results.

Empirically, FM-GANs demonstrate good performance on SSL classification tasks; however, the generated images from the generator are low-quality, *i.e*., the generator may create visually unrealistic images. Li *et al*. [16] realized that the generator and the discriminator in FM-GANs may not be optimal at the same time. Intuitively, assuming the generator

can create good samples, the discriminator should identify these samples as fake samples as well as predict the correct class for them. To address this problem, they proposed a three-player game, Triple-GAN, to simultaneously achieve superior classification results and obtain a good image generator. Meanwhile, Dai *et al.* [19] realized the same problem of the generator, but instead gave theoretical justifications of why using "bad" samples from the generator could boost SSL performance. Loosely speaking, they defined samples that form a complement set of the true data distribution in feature space as "bad" samples. Their model was called Bad GAN, which achieved better performance on multiple benchmark datasets compared to Triple-GAN.

Most recently, Li *et al.* [84] performed a comprehensive comparison between Triple-GAN and Bad GAN. They illustrated the distinct characteristics of the images the models generated, as well as each model's sensitivity to various amount of labeled data used for training. Furthermore, they showed that in the case of low amounts of labeled data, Bad GAN's performance decreased faster than Triple-GAN, and both models' performance were contingent on the selection of labeled samples; in other words, selecting non-representative samples would deteriorate the classification performance.

4.1.2 Related Work

Besides the aforementioned FM-GAN [113], Triple-GAN [16], and Bad GAN [19], several previous studies have also incorporated the idea of adversarial training in SSL. CatGAN [127] substituted the binary discriminator in standard GAN with a multi-class classifier and trained both the generator and discriminator using information theoretical criteria on unlabeled data. Virtual adversarial training (VAT) [97] effectively smoothed the classifier output distribution by seeking virtual adversarial samples. In adversarial learned inference [23], the inference network approximated the posterior of latent variables given true data in an unsupervised manner. Another line of work has focused on manifold regularization [7]. Kumar *et al.* [71] estimated the manifold gradients at input data points and added an additional regularization term to a GAN, which promoted invariance of the discriminator

to all directions in the data space. Lecouat *et al*. [75] achieved competitive results by performing manifold regularization using the approximate Laplacian norm that was easily computed within a GAN.

Apart from adversarial training, there have been other efforts in SSL recently. One class of the most successful algorithms in SSL are based on pseudo labels [53, 73, 105, 107, 132]. Pseudo labels are artificial labels generated by the model, which play the same role as labels of manually annotated data. Γ model [107] evaluated unlabelled data with and without noise, and applied a consistency cost between the two predictions. It assumed a dual role as a teacher and a student. The teacher generated targets of unlabeled data, which were then used to train a student. Since the model itself generated the targets, they could be incorrect. To alleviate the problem, Π model [73] added noise at the inference time, and consequently a noisy teacher could yield more accurate targets. Π model was further improved by Temporal Ensembling [73], which maintained an exponential moving average (EMA) prediction for each of the training examples. Consequently, the EMA prediction of each example was formed by an ensemble of the model's current version and those earlier versions that evaluated the same example. This ensembling improved the quality of the predictions, and using the predictions as teacher signals improved results. Mean Teacher [132] averaged model weights to form a target-generating teacher model. Unlike Temporal Ensembling, Mean Teacher worked with large datasets and on-line learning, which was able to improve the speed of learning and classification accuracy simultaneously.

Our proposed UGAN is mainly inspired by Triple-GAN and Bad GAN. These models can be used for complementary purposes. We restrict our discussion to GAN-based models for most of this chapter. Nevertheless, it has a connection with those "teacher" models, as will be seen in Section 4.2, our model provides a smart way to generate input-label pairs and use them as teaching signals to improve the SSL results.

4.1.3 Contributions

In this work, we present unified-GAN (UGAN), a semi-supervised learning framework that unifies both good and bad generated samples and takes advantage of them through adversarial training. Inspired by Triple-GAN and Bad GAN, we find that good and bad synthetic samples can be used for complementary purposes. Generated good image-label pairs can be used to train the classifier, while the bad samples can force the decision boundary to be between the data manifold of different classes. Hence, we leverage both good and bad generated samples in the proposed UGAN and achieve further performance improvement in SSL. Overall, our main contributions of this chapter are: 1) we propose a novel SSL framework, UGAN, which simultaneously trains a good and a bad generator through adversarial training and takes advantage of both generated samples to boost SSL performance; 2) we analyze our proposed UGAN, theoretically prove its global optimum, and additionally put UGAN in the Expectation-Maximization (EM) framework and validate its non-increasing divergence property; and 3) we do extensive experiments to show that UGAN can improve upon Triple-GAN and Bad GAN classification results in SSL, and show the effectiveness of the model with different amounts of labeled data.

4.1.4 Organization

The rest of the chapter is organized as follows. In Section 4.2, we present the problem definition and outline our approach to solve it. The experiments and discussions are presented in Section 4.3 followed by the limitations of the study in Section 4.4. Finally, Section 4.5 concludes the chapter.

4.2 Methods

To outline our approach, we consider the following SSL problem. Given a relatively small labeled set $(x_l, y_l) \sim p_l(x, y)$, where $y \in \{1, 2, \cdots, K\}$ is the label space for classification, and a large unlabeled set $x_u \sim p_u(x)$, the goal is to utilize the large amount of unlabeled data

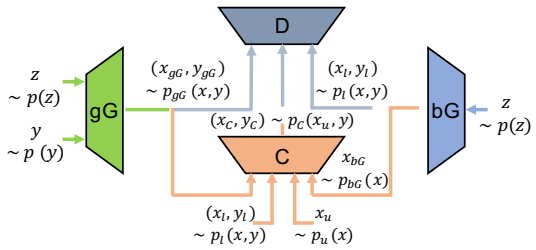

Figure 4.1: Network architecture of UGAN. UGAN consists of four components: 1) a bad generator, bG, generates "bad" samples; 2) two conditional networks, gG and C, that generate pseudo labels given real data, and pseudo data given real labels; and 3) a separate discriminator, D, that distinguishes the generated data-label pair from the real data-label pair. "CE" denotes the cross entropy loss for supervised learning, while "BCE" denotes the binary cross entropy loss that distinguish the real data and fake data generated by bG.

to predict the labels y of the unseen samples. Suppose the true data distribution is denoted as $p(x, y)$, we aim to obtain a classifier that can approximate the conditional distribution $p_C(y|x) \approx p(y|x)$. To achieve this, we will use an adversarial training process that enables the classifier to learn from both good and bad samples. Specifically, a good generator is able to generate good image-label pairs to train the classifier, while a bad generator generates samples that force the classifier's decision boundary between the data manifolds of different classes. As will be shown, our model takes advantage of both good and bad synthetic samples, and improves the SSL results in a wide range of labeled training data.

4.2.1 Adversarial Training Process with Four Players

Our model consists of four parts: 1) a good generator, gG, that characterizes the conditional distribution $p_{gG}(x|y) \approx p(x|y)$; 2) a bad generator, bG, that takes in a latent vector z and outputs "bad" samples [19]; 3) a classifier, C, that characterizes the conditional distribution $p_c(y|x) \approx p(y|x)$; and 4) a discriminator, D, that distinguishes whether a pair of data (x, y) comes from the true distribution $p(x, y)$ or not. All the components are parameterized as neural networks, as shown in Figure 4.2(a).

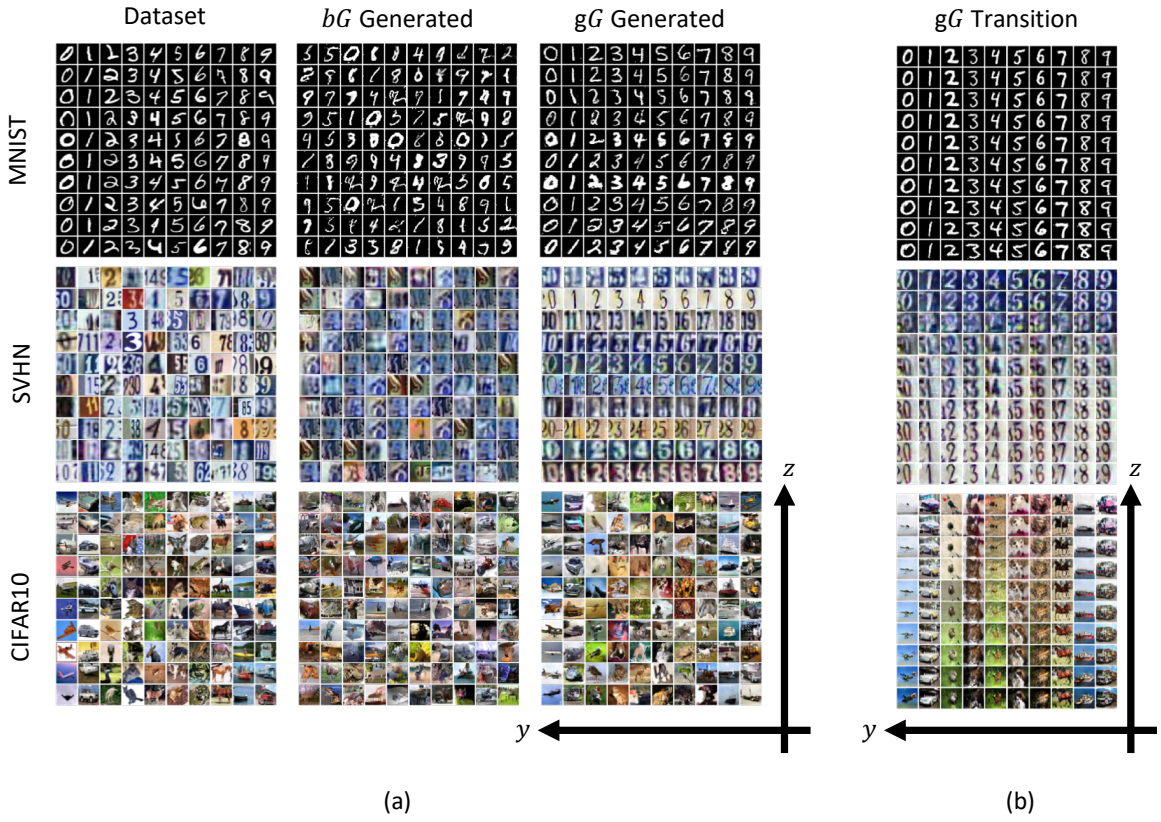

Figure 4.2: (a) Left: randomly selected data from datasets; mid: bG generated images; right: gG generated images sampled by varying the class label y in the horizontal axis and the latent vectors z in the vertical axis. (b) Class-conditional latent space interpolation. The vertical axis is the direction for latent vector interpolation, while the horizontal axis for varying the class labels.

We assume that the samples from both real data $p(x)$ and real label $p(y)$ can be easily obtained.[1] In our model, gG produces a pseudo input-label pair by first drawing $y \sim p(y)$ and latent vector $z \sim p(z)$ (we use a uniform distribution for z in our experiments), and then generating $x_{gG} \sim p_{gG}(x|y,z)$. bG generates bad samples by transforming the latent vector

[1]In semi-supervised learning, $p(x)$ is the empirical distribution of inputs and $p(y)$ is assumed same to the distribution of labels on labeled data, which is uniform in our experiments.

$z \sim p(z)$ as in a traditional GAN to obtain $x_{bG} \sim p_{bG}(x|z)$. C takes in four different types of samples (*i.e.*, labeled data, unlabeled data, samples from gG, and samples from bG) and produces pseudo labels y for them following the conditional distribution $p_C(y|x)$. For the labeled data x_l, and the gG generated samples x_{gG}, we expect C to put them into the right class (*i.e.*, either the class y_l of the labeled data x_l, or the conditional labels y based on which x_{gG} are generated). For the generated samples from bG $x_{bG} \sim p_{bG}(x|z)$, and unlabeled data $x_u \sim p_u(x)$, we expect C to put them into the $(K+1)$th class (*i.e.* the "fake" class) and one of the K classes of real data, respectively. Due to the fact that the softmax layer is over-parameterized, we can still model C with K neurons at the output layer by modifying the loss function (see details in Appendix C.1). D accepts the input-label pairs generated by both C $(x_C, y_C) \sim p(x_u)p_C(y|x_u)$, and gG $(x_{gG}, y_{gG}) \sim p(y)p_{gG}(x|y)$, and the pairs from the labeled data distribution $(x_l, y_l) \sim p_l(x, y)$ for judgement. D treats the labeled data pairs as positive samples, while the pairs from both gG and C as negative. We refer to the loss function of gG as[2]

$$L_{gG} = \mathbb{E}_{x,y\sim p_{gG}(x,y)}[\log(1 - p_D(x, y)] \tag{4.1}$$

The loss function of bG is

$$\begin{aligned} L_{bG} = &-\mathcal{H}(p_{bG}(x)) \\ &+ \left\| \mathbb{E}_{x\sim p_u(x)}(\boldsymbol{f}(x)) - \mathbb{E}_{x\sim p_{bG}(x)}(\boldsymbol{f}(x)) \right\|_2^2 \end{aligned} \tag{4.2}$$

where $-\mathcal{H}(p_{bG}(x))$ measures the negative entropy of bG generated samples. $-\mathcal{H}(p_{bG}(x))$ is used to avoid collapsing while increasing the coverage of bG. The second term is feature matching loss, where $\boldsymbol{f}(x)$ denotes a feature map of an intermediate layer of C. D's loss

[2]In practice, we use $L_{gG} = -\mathbb{E}_{x,y\sim p_{gG}(x,y)}[\log(p_D(x, y)]$ to ease the training process [37].

function becomes

$$\begin{aligned} L_D = &- \mathbb{E}_{x,y\sim p_l(x,y)}[\log(p_D(x,y)] \\ &- \frac{1}{2}\mathbb{E}_{x,y\sim p_{gG}(x,y)}[\log(1-p_D(x,y)] \\ &- \frac{1}{2}\mathbb{E}_{x,y\sim p_C(x,y\leqslant K)}[\log(1-p_D(x,y)] \end{aligned} \tag{4.3}$$

where D treats the labeled data as positive samples, and the pseudo input-label pairs from both gG and C as negative samples. Finally, the loss function of C consists of four components,

$$\begin{aligned} L_{C_1} &= -\mathbb{E}_{x,y\sim p_l(x,y)}[\log(p_C(y|x,y\leqslant K)] \\ L_{C_2} &= -\mathbb{E}_{x,y\sim p_{gG}(x,y)}[\log(p_C(y|x,y\leqslant K)] \\ L_{C_3} &= -\mathbb{E}_{x\sim p_u(x)}[\log(1-p_C(y=K+1|x)] \\ L_{C_4} &= -\mathbb{E}_{x\sim p_{bG}(x)}[\log(p_C(y=K+1|x)] \end{aligned} \tag{4.4}$$

and the total loss for C is

$$L_C = L_{C_1} + \lambda_0 L_{C_2} + \lambda_1 L_{C_3} + \lambda_2 L_{C_4} \tag{4.5}$$

where L_{C_1} and L_{C_2} denote the cross entropy loss for labeled and gG generated samples, L_{C_3} forces C to put the unlabeled data into real classes, while L_{C_4} forces C to put the bG generated samples into the "fake" class. $\lambda_{0,1,2}$ is a hyperparameter used to balance each loss component.

The model defined by (4.1)-(4.5) achieves its equilibrium if and only if $p(x,y) = p_{gG}(x,y) = p_C(x,y\leqslant K)$. In other words, incorporating the bad samples does not change the equilibrium point of Triple-GAN (see Section 4.2.2.1). Our model consists of three adversarial parts: 1) gG tries to fool D by generating realistic images conditioned on label y; 2) C tries to fool D by generating good labels for unlabeled images; and 3) bG tries to fool C by generating images that are close to the data manifold. At convergence, D cannot distinguish both $p_{gG}(x,y)$ and $p_C(x,y)$ from the true data distribution $p(x,y)$, which indicates that we have obtained both a good gG and a good C. Bad samples from bG accelerate this process and improve the generalization of C.

One key problem of SSL is the limited amount of labeled data. A powerful D may memorize the empirical distribution of the labeled data, and reject other types of samples from the true data distribution. Limited labeled data also restricts gG to explore a larger space of the true data distribution. To address this problem, we adopt the practical techniques in Li *et al.* [16]. We generate pseudo labels through C for some unlabeled data and use these pairs as positive samples of D. This introduces some bias to the target distribution of D, but using the EM framework to analyze the training procedure (see Section 4.2.2.2), we are able to prove the rationality of this choice. Moreover, since C converges quickly, this operation provides a way to enable gG to explore a much larger data manifold that includes both the labeled and unlabeled data information. As illustrated in Figure 4.2 (b), C is able to provide pseudo labels for the unlabeled data, while D will judge if the pseudo labels are reliable or not. This in return will affect the evolution of gG that will take advantage of the unlabeled data to generate good images. Generated good image-label pairs that implicitly contain unlabeled data information will eventually benefit C. This works extremely well for relatively simple datasets like MNIST, and under the circumstance where only an extremely low amount of labeled data is available.

4.2.2 Theoretical Analysis

We now give theoretical justification for our four-player game based on the loss functions as mentioned above. We mainly focus on two important properties of our model: 1) the global optimum of the game is the true distribution, which satisfies $p(x, y) = p_{gG}(x, y) = p_C(x, y|y \leqslant K)$; and 2) the KL divergence between the conditional density of C and the true density, $\mathrm{KL}(p(y|x)||p_C(y|x, y \leqslant K))$, is non-increasing after each iteration when we assume the maximum likelihood estimate (MLE) of C is obtained. A detailed proof of these properties is provided in Appendix C.3.

4.2.2.1 Global Optimum

We first show that the optimal D balances between the true data distribution and the mixture distribution defined by C and gG, as summarized in Lemma 1.

Lemma 1. *For any fixed C and gG, the optimal D of the game defined by loss functions (4.1)-(4.4) is*

$$D^*_{C,gG,bG}(x,y) = \frac{p_l(x,y)}{p_l(x,y) + p_{\frac{1}{2}}(x,y)}, \tag{4.6}$$

where $p_{\frac{1}{2}}(x,y) = \frac{1}{2}p_{gG}(x,y) + \frac{1}{2}p_C(x,y|y \leqslant K)$.

Given $D^*_{C,gG,bG}$, we can plug the optimal D^* into (4.3) and get a value function $V(C,gG,bG)$.

$$\begin{aligned}
V_{C,gG,bG}(x,y) =& -\mathbb{E}_{x,y\sim p_l(x,y)}[\log(p_{D^*}(x,y)] \\
&-\frac{1}{2}\mathbb{E}_{x,y\sim p_{gG}(x,y)}[\log(1-p_{D^*}(x,y)] \\
&-\frac{1}{2}\mathbb{E}_{x,y\sim p_C(x,y\leqslant K)}[\log(1-p_{D^*}(x,y)] \\
=& -\mathbb{E}_{x,y\sim p_l(x,y)}[\log(\frac{p_l}{p_l+p_{1/2}})] \\
&-\frac{1}{2}\mathbb{E}_{x,y\sim p_{gG}(x,y)}[\log(\frac{p_{1/2}}{p_l+p_{1/2}})] \\
&-\frac{1}{2}\mathbb{E}_{x,y\sim p_C(x,y\leqslant K)}[\log(\frac{p_{1/2}}{p_l+p_{1/2}})]
\end{aligned} \tag{4.7}$$

Now the left problem is to maximize the $V(C,gG,bG)$, so that gG and C confuse D most. For that, we have the following theorem:

Theorem 2. *The global maximum of $V(C,gG,bG)$ is achieved only when $p_l(x,y) = p_{gG}(x,y) = p_C(x,y|y \leqslant K)$.*

From (4.7), it is easy to see the global maximum is achieved if and only if $p_l(x,y) = p_{1/2}(x,y)$. By introducing the cross-entropy loss in (4.4) L_{C1}, we enforce $p_C(x,y|y \leqslant K) = p_l(x,y)$. Therefore, the global optimality will achieve if and only if $p_l(x,y) = p_{gG}(x,y) = p_C(x,y|y \leqslant K)$. (See more details in Appendix C.3)

We now consider the case for $p_C(y = K+1|x)$ with the following Corollary 2.1.

Corollary 2.1. *The optimal classifier C will have $p_C(y = K+1|x \sim p_u(x)) = 0$ and $p_C(y = K+1|x \sim p_{bG}(x)) = 1$.*

Corollary 2.1 indicates that optimal C will put bG generated images into $K+1$ class (i.e., "fake" class), while put unlabeled data into real classes.

4.2.2.2 Non-increasing Divergence Property

Our goal is to estimate the conditional distribution $p(y|x)$ with a parameterized C modeled as $p_\theta(y|x, y \leqslant K)$. The objective function can be written as minimizing $\mathrm{KL}(p(y|x)||p_\theta(y|x, y \leqslant K))$. In the SSL setting, we only have part of the labels y, so we can thus rewrite the problem as minimizing $\mathrm{KL}(p(y_l|x)||p_\theta(y_l|x, y \leqslant K))$. One natural way to facilitate the model performance is using the EM algorithm to first infer the label of x_u and then update based on the complete data [101]. In our four-player game, in addition to the predicted label y_u from unlabelled data x_u, we further introduce (x_{gG}, y_{gG}) pairs from gG as latent variables, denoted as $Z = \{x_{gG}, y_{gG}, y_u\}$. We then interpret our mechanism from a variational view of the EM algorithm to illustrate the non-increasing property of the KL divergence.

Property I. Chain rule of KL divergence:

$$\begin{aligned}\mathrm{KL}(P(X,Z)||P_\theta(X,Z)) = \mathrm{KL}(P(X)||P_\theta(X)) \\ +\mathbb{E}_{x\sim P(X)}[\mathrm{KL}(P(Z|x)||P_\theta(Z|x))].\end{aligned} \tag{4.8}$$

By ***Property I***, we can rewrite our objective function as:

$$\begin{aligned}\min_\theta \mathrm{KL}(p(y_l|x)||p_\theta(y_l|x, y \leqslant K)) = \\ \min_\theta \min_{p(Z|x)} \mathrm{KL}(p(y_l, Z|x)||p_\theta(y_l, Z|x, y \leqslant K)),\end{aligned} \tag{4.9}$$

which is an iterative minimization procedure. Following the EM algorithm, we have an *E-step* and an *M-step* in UGAN. More specifically, for the *E-step* at the sth iteration, given parameters θ_s of C, we have:

$$\begin{aligned}p(Z|x) = p_{\theta_s}(Z|x) = \\ p_{gG}(x_{gG}, y_{gG}|x_u, x_l, y_u, y_l)p_{\theta_s}(y_u|x_u),\end{aligned} \tag{4.10}$$

which indicates the procedure that C first predicts labels for unlabelled data, and then sends them to D and gG to generate good pseudo pairs (x_{gG}, y_{gG}). After gathering the latent variables, the *M-step* is:

$$\begin{aligned}\theta_{s+1} &= \text{argmin}_\theta \text{KL}(p(y_l, Z|x)||p_\theta(y_l, Z|x, y \leqslant K)) \\ &= \text{argmax}_\theta \mathbb{E}_{(y_l, Z|x)\sim f}[\log p_\theta(y_l, Z|x, y \leqslant K)],\end{aligned} \tag{4.11}$$

where $f = p_{\theta_s}(Z|x_u)p_l(y_l|x_l)$. This will result in θ_{s+1} being the MLE based on the data at current iteration s. By applying the EM mechanism, we can inherit its non-increasing property which is stated in the following Corollary 2.2.

Corollary 2.2. *If applying the iterative procedure described in (4.10) and (4.11), and the exact maximization can be obtained at (4.11) for each iteration, then*

$$\begin{aligned}&KL(p(y_l|x)||p_{\theta_{s+1}}(y_l|x, y \leqslant K)) \leqslant \\ &KL(p(y_l|x)||p_{\theta_s}(y_l|x, y \leqslant K))\end{aligned} \tag{4.12}$$

The non-increasing property guarantees that our classifier will be improved after each iteration under the ideal situation. Though we make some approximations during the training process in practice, it still provides us a high-level justification on why the algorithm should work.

4.3 Experiments and Discussion

We now present UGAN's performance on MNIST [77], SVHN [99], and CIFAR10 [69] datasets (see details of datasets in Appendix C.4). We implement our model based on Tensorflow 1.10 [31] and optimize it on NVIDIA Titan X GPUs. The detailed architecture can be found in Appendix C.5. The gG generated images are not applied until the number of epochs reaches a threshold such that gG can generate reliable image-lable pairs. For MNIST and SVHN, we choose 200, while for CIFAR10 we choose 400. Batch size is an important parameter that affects model performance [84]. In our experiments, we use 50 for bG on MNIST and SVHN, 25 for bG on CIFAR10. For gG, we fix batch size as 100. All of

the other hyperparameters including relative weights and parameters in Adam [62] are fixed according to [16, 19, 113] across all of the experiments.

4.3.1 Classification

We report our classification accuracy, along with other GAN-based SSL methods, on benchmark datasets in Table 4.1. Our results show that UGAN consistently improves performance, and achieves the best results on all of the datasets without the use of data augmentation, such as rotation, flip, *etc*.

To understand our model's behavior over different numbers of labeled data, we re-implemented Triple-GAN and Bad GAN, and performed an extensive investigation by varying the amount of labeled data. Following common practice, this was done by omitting different amounts of the underlying labeled dataset [104, 112, 113, 132]. The labeled data used for training were randomly selected stratified samples unless otherwise specified. For fair comparison, we used the same network architecture for each component in all models (see Appendix C.5). Table 4.2 shows the results of the experiments on MNIST. The similarity of our results to those reported in the original papers suggests that our reproduced models are accurate instantiations of Triple-GAN and Bad GAN. We observe that with a medium amount of labeled data (e.g., MNIST $n = 100$), Bad GAN performs better than Triple-GAN. However, with smaller amounts of labeled data, Triple-GAN performs better, which demonstrates that it is less sensitive to the amount of labeled data than Bad GAN. UGAN inherits the good properties from both of them, resulting in a constant improvement across all cases (see results on SVHN and CIFAR10 in Appendix C.6). Another interesting observation is that the selection of labeled data plays a crucial role in the low-labeled data regime, that is, selecting representative labeled data with which to train is the key to achieving good performance. This issue is further discussed in Appendix C.7.

To further validate that our model significantly improves the baseline model, we have performed Welch's t-test. We found that our model significantly improves Triple-GAN and Bad-GAN with a maximum p-value in order of 1e-5 for both SVHN and CIFAR10 datasets.

For the MNIST dataset, we have found that for some results, our model's performance is not significantly different form the literature, such as the Bad-GAN case with 100 labeled samples. This is because the MNIST classification task is a relatively easy task, and the previous study has already achieved very good performance. The small p-values for SVHN and CIFAR10 have demonstrated that our model improves the baseline model significantly on more complex datasets.

Table 4.1: Comparison with state-of-the-art methods on three benchmark datasets. Only methods without data augmentation are included. Results are averaged over 10 runs and shown in terms of mean accuracy ± standard deviation.

Methods	MNIST $n = 100$	SVHN $n = 1000$	CIFAR10 $n = 4000$
CatGAN [127]	$98.09 \pm 0.1\%$	-	$80.42 \pm 0.46\%$
ALI [23]	-	$92.58 \pm 0.65\%$	82.01 ± 1.62
VAT [97]	98.64%	93.17%	85.13%
Π Model [73]	-	$94.57 \pm 0.25\%$	$83.45 \pm 0.29\%$
Γ Model [107]	$99.11 \pm 0.50\%$	-	$79.40 \pm 0.47\%$
Mean Teacher [132]	-	$96.05 \pm 0.19\%$	$84.27 \pm 0.31\%$
FM-GAN [113]	$99.07 \pm 0.07\%$	$91.89 \pm 1.3\%$	$81.37 \pm 2.32\%$
Triple-GAN [16]	$99.09 \pm 0.58\%$	$94.23 \pm 0.17\%$	$83.01 \pm 0.36\%$
Bad-GAN [19]	$99.21 \pm 0.10\%$	$95.75 \pm 0.03\%$	$85.59 \pm 0.30\%$
UGAN	$\mathbf{99.21 \pm 0.08\%}$	$\mathbf{96.49 \pm 0.09\%}$	$\mathbf{85.66 \pm 0.06\%}$

4.3.2 Image Generation

UGAN is able to train a gG and a bG simultaneously (see an evolution of the generated images in Appendix C.8). In Figure 4.2 (a), we show the images generated by gG and bG after training. Our gG is able to generate clear images and meaningful samples conditioned on class labels, while bG generates "bad" images that look like a fusion of samples from different classes. We quantitatively evaluate generated samples on CIFAR10 via the inception score

Table 4.2: Test accuracy on semi-supervised MNIST. Results are averaged over 10 runs and shown in terms of mean accuracy ± standard deviation. $*$ denotes hand selection of labeled data. † denotes our implementation of the model.

Model	Test accuracy for a given number of labeled samples			
	20	50	100	200
FM-GAN [113]	$83.23 \pm 4.52\%$	$97.79 \pm 1.36\%$	$99.07 \pm 0.07\%$	$99.10 \pm 0.04\%$
Bad GAN [19]	-	-	$99.21 \pm 0.10\%$	-
Triple-GAN [16]	$95.19 \pm 4.95\%$	$98.44 \pm 0.72\%$	$99.09 \pm 0.58\%$	$99.33 \pm 0.16\%$
Bad GAN†	$88.38 \pm 3.08\%$*	$96.24 \pm 0.16\%$	$99.17 \pm 0.03\%$	$99.20 \pm 0.03\%$
Triple-GAN†	$95.93 \pm 4.45\%$*	$98.68 \pm 1.12\%$	$99.07 \pm 0.46\%$	$99.17 \pm 0.08\%$
UGAN	$\mathbf{97.34 \pm 6.86\%}$*	$\mathbf{98.92 \pm 0.13\%}$	$\mathbf{99.21 \pm 0.08\%}$	$\mathbf{99.35 \pm 0.05\%}$

following [113]. The value of gG generated samples is 4.19 ± 0.07, while that of bG generated samples is 3.31 ± 0.02. In addition, gG retains Triple-GAN's advantage in that it is able to disentangle classes and styles. In Figure 4.2(a), the gG generated images are sampled by varying the class label y in the horizontal axis and the latent vectors z in the vertical axis. The latent vector z encodes meaningful physical appearances, such as scale, intensity, orientation, color, *etc.*, while the label y controls the semantics of the generated images. Furthermore, gG can transition smoothly from one style to another with different visual factors without losing the label information as shown in Figure 4.2 (b). This demonstrates that gG can learn meaningful latent representations instead of simply memorizing the training data.

4.3.3 Hyper-parameters Sensitivity Analysis

We perform hyper-parameter sensitivity analysis along with some network architecture effects. We discover that the hyper-parameters used in Triple-GAN and Bad GAN are also good for UGAN. In fact, aside from batch size, we use the same hyper-parameters across all three datasets, and consistently achieve good results. UGAN is not sensitive to the learning rate due to the usage of Adam optimization as shown in Table 4.3. However, UGAN is quite

sensitive to the batch size in training. Our results indicate the noise induced by mini-batch benefits the bG, while it hurts the gG capability to model the true data distribution. We also find that a weight-norm layer is important to ease GAN's training. UGAN doesn't usually converge when the layer is taken out. A smaller network architecture of C would not result in a significant drop in the performance. We use a C with filter size {32, 64, 96} and get 96.27% on SVHN n = 4000. For details on how our hyper-parameters sensitivity analysis is performed, we refer readers to Appendix C.9.

Table 4.3: Initial Learning Rate Effect on Model Performance. The experiments are done on MNIST $n = 100$. Despite the differences of the training loss in the initial stage, the final results are not significant different after training 400 epochs.

Learning Rate	$lr = 1e-2$	$lr = 1e-3$	$lr = 5e-4$	$lr = 3e-4$
Accuracy	99.13%	99.18%	99.24%	99.18%

4.3.4 Effectiveness of Good and Bad Generators

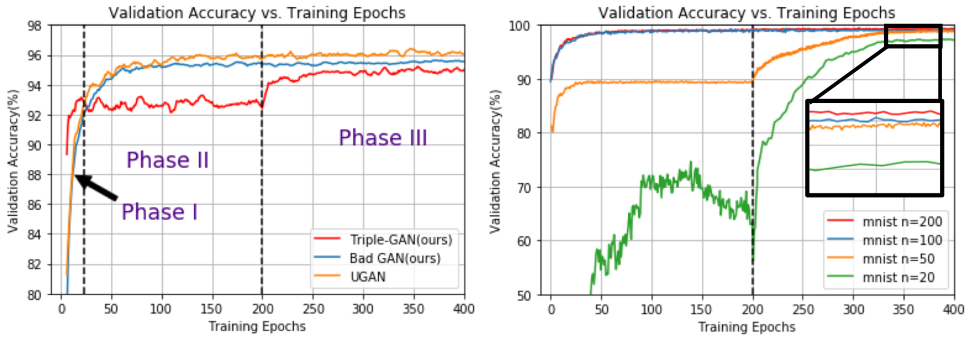

Figure 4.3: (a) Comparison of Validation Accuracy vs. Training Epochs on our implemented Triple-GAN, Bad GAN, and UGAN. The experiments are performed on SVHN $n = 1000$. (b) UGAN Validation Accuracy vs. Training Epochs under various amounts of labeled data on MNIST.

As discussed in Section 4.3.1, UGAN achieves consistent improvement across all the cases

due to inheriting the best properties of Triple-GAN and Bad GAN. In Figure 4.3 (a), we demonstrate a comparison of Validation Accuracy vs. Training Epochs for our implemented Triple-GAN, Bad GAN, and UGAN on SVHN $n = 1000$. Note that for Triple-GAN, we trained it to 1000 epochs, but only show the first 400 epoch in the figure. Qualitatively, we observe three separate training phases:

1. In **Phase I**, the performance of Bad GAN and UGAN are worse than Triple-GAN. We speculate this is due to the fact that Triple-GAN C deals with a classification of K classes, while Bad-GAN and UGAN, C deal with $K + 1$ classes.

2. In **Phase II**, Bad GAN and UGAN start to surpass Triple-GAN, which indicates bG generated samples start to exert an effect on the classification boundary. UGAN also performs better than Bad GAN in this phase thanks to the adversarial game that requires C to produce reliable pseudo labels for unlabeled data to fool D.

3. In **Phase III**, we start to use gG generated samples to train C. UGAN surpasses both Triple-GAN and Bad GAN by a clear margin. From the perspective of C, gG generates samples that are used to complement the lack of training data in SSL, bG generated samples are used to force the decision boundary to lie in the correct place, and D requires C to keep moving itself toward the true data distribution $p(x)p_C(y|x, y \leqslant K) \approx p(x, y)$. All of these factors contribute to the final performance of UGAN.

Similar observations can also be found in Appendix C.10 on MNIST and CIFAR10. Moreover, we hypothesize that for fewer labeled data, gG plays an important role, as gG is able to model the class-aware data distribution under weak supervision and use them to complement the lack of the training samples. For larger labeled data, bG plays a more important role by generating complementary samples and forcing the decision boundary to lie between the data manifolds of different classes. Empirically, we show our model's validation accuracy under various amounts of labeled data on MNIST in Figure 4.3 (b). As can be seen, when we push the number of labeled data to extremely low numbers, the training curve becomes more like that in Triple-GAN i.e., a bump is shown clearly at epoch = 200 when we

start to use gG generated samples to train C. However, we do not find a similar transition on SVHN and CIFAR10 (see Appendix C.10). One possible explanation is that when we use too few labeled data, gG fails to model the conditional distribution due to the complexity of SVHN and CIFAR10. Note that we only used traditional techniques for training the GAN. With recent advances in generating high quality images using GANs [9, 90, 96], our model may be able to achieve further performance improvements on more complex datasets with even fewer labeled data.

4.4 Limitations of the Study

Here, we discuss some limitations of our work and provide potential research directions that could help address these limitations.

We note that we assume the marginal distribution $p(y)$ to be uniform, which is easy to sample for the generation process. However, it is not always true for other applications where $p(y)$ is no longer uniform. In these cases, we expect that the non-uniform label distribution affects both good sample generation and classification. For good sample generation, the generator will have difficulty capturing features for the minority class. For classification, the classifier tends to cheat by always predicting the majority class. These problems are expected to be more severe when the dataset is highly skewed. One potential future research direction is to investigate how a non-uniform label distribution will affect our model and how common data balancing methods such as upsampling and data augmentation can provide help to it.

Another area for potential investigation is to generate high-resolution, large-scale images, so that our model can be used in more complex scenarios. In this chapter, we have only applied the model to relatively simple datasets with less complexity, such as MNIST (28*28), SVHN (32*32) and CIFAR10 (32*32). Part of the reason is that the model in the current form is not able to generate reliable image-label pairs on large-scale. With the advancement in high-resolution large-scale image generation using GAN recently, we expect that our model

will be able to applied to much more complex scenarios such as ImageNet classification, and pixel-wise segmentation.

4.5 Conclusions

We have presented unified-GAN (UGAN), a new GAN framework for semi-supervised learning. By learning from good and bad samples through adversarial training, we have demonstrated that our model performs better on image classification tasks across several benchmark datasets and under a range of labeled training data. We envision that UGAN can be used in a variety of scenarios, such as healthcare, where obtaining labeled data can be expensive and time-consuming. We also consider adapting UGAN to other types of data such as text (e.g. improving SSL text categorization performance for 20 newsgroups, Reuters, NYTimes, Wiki, PubMed *etc.*).

CHAPTER 5

High Resolution Histopathology Image Generation and Segmentation through Adversarial Training

5.1 Introduction

5.1.1 Motivation

Review of histopathology slides is important in medical diagnosis and treatment planning. It requires accurate quantitative analysis such as morphological feature extraction and cancer grading. Good segmentation may pave the way for these analyses and increase their reproducibility [82]. Recently, deep learning has brought significant improvement to semantic segmentation in medical image analysis. However, its performance typically relies on large annotated datasets [88], and thus segmentation of histopathology images remains a challenge given the relative paucity of annotations. The images resulting from digitized histopathology slides are inherently high resolution (high-res), and obtaining large amounts of annotated data is laborious. Moreover, histopathological features in the images vary widely for different cancer grades, making them difficult to segment at a granular level.

Recently, generative adversarial networks (GANs) have been rapidly adopted by the medical imaging community [147]. GANs have shown promise in data augmentation as they are able to synthesize high quality data to help overcome privacy issues and tackle the insufficiency of training data. However, most studies have focused on relatively low-resolution, small-scale images, such as CT and MRI [3, 148]. The few GAN-based methods applied on pathology image synthesis focus only on cell-level feature representation [72, 93, 121]. High-res histopathology images contain diverse descriptors, and require GANs to preserve spatial

consistency at a large scale, which has posed a challenge for their synthesis. In this work, we propose a high-res large-scale histopathology image generation and segmentation framework through adversarial training. By setting up a dedicated multi-scale/pyramid training scheme, we are able to synthesize realistic histopathology images conditioned on semantic masks and use the synthesized images to train a segmentation network end-to-end in both fully-supervised and semi-supervised scenarios. To the best of our knowledge, our approach is the first conditional generation model for high-res histopathology images and the first approach to use image synthesis for high-res histopathology image augmentation. Extensive experiments have been conducted to show the effectiveness of our proposed method in both image generation and semi-supervised segmentation. Detailed analysis is also provided to demonstrate how the multi-scale/pyramid structure and synthetic data augmentation each contribute to the model's performance.

5.1.2 Related Works

5.1.2.1 Conditional GAN for Image Synthesis

Many researchers have leveraged conditional adversarial learning for image synthesis (also known as image-to-image translation), whose goal is to generate images based upon the conditional input. For example, in natural images, pix2pix framework [54] used image-conditional GANs for different applications, such as generating cats from user sketches and transforming Google maps to satellite views. On top of it, pix2pixHD [136] proposed a multi-layer discriminator to synthesize high-resolution photo-realistic imagery without any hand-crafted losses or pre-trained networks. SinGAN [116], on the other hand, introduced a pyramid of fully convolutional GANs, each responsible for learning the patch distribution at a different scale of a single image. In our proposed models, we incorporate the ideas of multi-layer discriminator and construct the image generation in a pyramid fashion.

Medical images can also be generated by implementing constraints on segmentation maps. Guibas *et al*. and Costa *et al*. proposed a two-stage process that first trained a segmentation

network to produce the vessel geometry, and then used the produced masks to synthesize fundi images [17, 41]. Mok *et al*. proposed a coarse-to-fine network to generate brain MR images conditioned on a segmentation mask [98]. Senaras *et al*. proposed a conditional GAN model for generating pathology images conditioned on nuclei segmentation masks [114]. Unlike the above models, our model works on high-res gland-level histopathology image generation, and we further leverage the synthetic images to train the segmentation tasks end-to-end in both supervised and semi-supervised settings.

5.1.2.2 GAN for Segmentation

GANs have been used for segmentation tasks in medical images. In these cases, the discriminator can be regarded as a regulator and the adversarial loss can be viewed as a similarity measure between the segmented outputs and the annotated ground truth. Kamnitsas *et al*. proposed an unsupervised domain adaptation model using adversarial neural networks to train a segmentation task on brain MR datasets [60]. Yang *et al*. achieved cross-modality domain adaptation, *i.e*. between CT and MRI images, via disentangled representations using adversarial training [145]. Xue *et al*. used a multi-scale L_1 loss as a similarity measure in the BRATS challenges [143]. Li *et al*. introduced an auxiliary classifier to regularize both the discriminator and the segmenter for fluorescent images [85]. Mahmood *et al*. demonstrates a nuclei segmentation methods across different organs using deep adversarial training [93]. Unlike the above models, our model consists of three components: a generator that can generate good images conditioned on the mask, a segmenter that can segment the input histopathological images, and a discriminator that distinguishes the ground-truth image-mask pairs from the pseudo image-mask pairs. The three network components forms two adversarial games in training: one is between the generator and the discriminator that helps the generator to synthesize realistic images to compensate for the limited data size; and the other is between the segmenter and the discriminator to help regularize the segmenter, so that the segmenter can output better masks to deceive the discriminator. Compared to the above models, our method provides two advantages in achieving better segmentation results:

(1) the conditional generated images by the generator will be used to compensate the lack of training data (2) the adversarial game between segmenter and discriminator will regularize the model to learn the image-mask distribution.

5.1.2.3 GAN for Semi-supervised Learning

Several studies have adopted semi-supervised learning (SSL) training schemes using GANs in medical image classification problems. Madani *et al*. and Lecouat *et al*. found that an SSL-GAN can achieve comparable performance with traditional convolutional neural networks with less data in chest abnormality classification, retinal vessel classification, and cardiac disease diagnosis [75, 92]. Most of the other works that used GANs to generate new training samples applied a two stage process, with the first stage trained to augment the images and the second stage trained to perform a classification task. In contrast, our approach utilizes a single model that is capable of performing conditional synthesis and uses it to improve the downstream segmentation task simultaneously. Furthermore, there is limited research on segmentation in SSL on histopathology images. Zhang *et al*. proposed to use both annotated and unannotated images in a segmentation task, where the unannotated images are used to compute the segmentation masks to confuse the discriminator [149]. Bulten *et al*. used a semi-automatic segmentation method to generated semantic mask and grade prostate biopsies [11]. To the best of our knowledge, we are the first to explore GAN data augmentation effectiveness for segmentation in an SSL framework on histopathology images.

5.1.3 Contributions

The main contributions of this study are twofold. First, by using a pyramid generation scheme, we are able to generate large-scale histopathological images up to 1024x1024 at high resolution (20x). Compared to the state-of-the-art pathology synthesis methods, which generate images up to 256x256 allowing for only limited context such as simple nuclei [93,114], our generation allows to incorporate richer context such as gland structures and nuclei details

that are useful for precise diagnosis. Second, the generation is based upon a conditional method, which produces good image-mask pairs. These image-mask pairs can be used to compensate the lack of data points in training segmentation models. We demonstrate the effectiveness of our method in segmentation tasks and analyze how it performs differently in supervised and semi-supervised settings.

5.1.4 Organization

The rest of this chapter is organized as follows: we first discuss the details of our proposed model in Section 5.2. The datasets used in our experiments and experimental results are discussed in Section 5.3. We also discuss the limitations of our work and provide directions for possible future work in Section 5.4. Finally, conclusions are drawn in Section 5.5.

5.2 Methods

Our goal is to synthesize realistic histopathology images x based on an arbitrary semantic mask y, so that (x, y) can be used to compensate for a small data size when training a segmentation network. Image synthesis for data augmentation using GAN is not new, but it is not widely used in histopathology analysis because generating images with fine details is difficult. Synthesizing images for gland segmentation poses even more challenges, as the generated images have to preserve both global gland structures and finer nuclear details based on the masks on a large scale. To overcome these problems, we design the generation and segmentation networks using pyramid structures. Subsequently, we show that the synthesized image-mask pairs can be used to boost segmentation performance, especially in the semi-supervised scenario seen in Section 5.3.

5.2.1 Generation

To synthesize high-res, large-scale histopathology images conditioned on semantic masks, our model must capture the statistics of complex image features at different scales. We wish

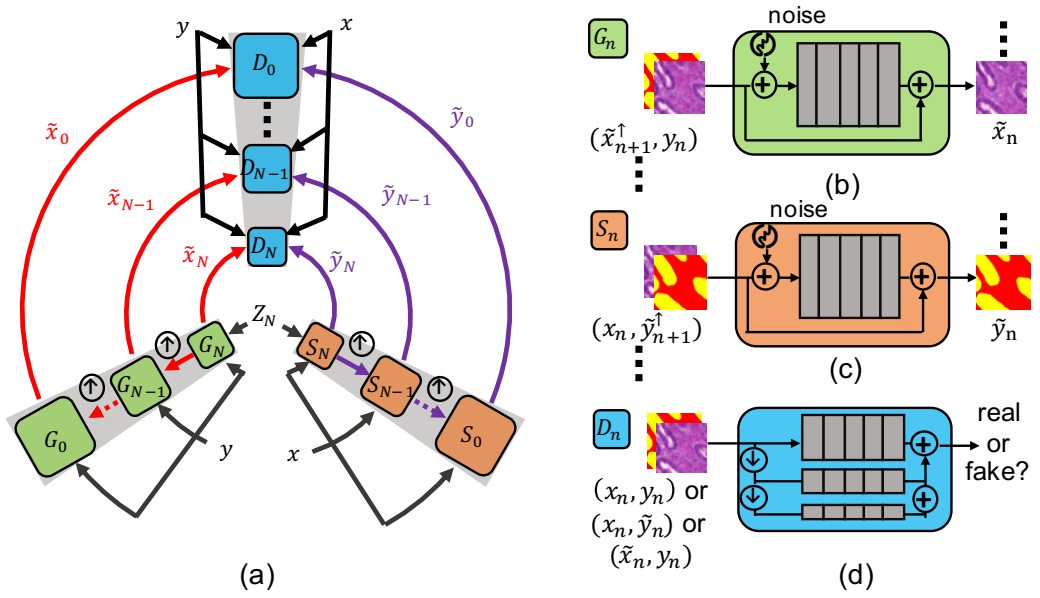

Figure 5.1: Schematics of our approach. (a) A pyramid model that consists of a Generator G_n, a Segmenter S_n, and a Discriminator D_n at each scale. G_n synthesizes image based on mask y_n and lower-scale generation $\tilde{x}_{n+1}$; S_n segments image based on image x and lower-scale segmentation $\tilde{y}_{n+1}$; D_n enforces image-mask pairs from both G_n and S_n to match the real distribution. Once G_n achieves good results, we can use the synthetic data to train S_n. This path has been omitted in the figure for simplicity. Note that noise is injected to G_n and S_n, which has also been omitted in the figure. (b) Illustration of the generator G_n. Each generator G_n attempts to generate realistic images $\tilde{x}_n$ conditioned on y_n and the previous generated images $\tilde{x}_{n+1}$. (c) Illustration of the segmenter S_n. S_n is symmetric with G_n as it conditions on the input image x_n and lower-scale segment results $\tilde{y}^{\uparrow}_{n+1}$, and attempts to segment the x_n. (d) Illustration of the discriminator D_n. D_n takes in image-mask pairs as input, and differentiates whether they are real (x_n, y_n) or fake $(\tilde{x}_n, y_n)$ from the generator or the segmenter. To differentiate large-scale high-res real and synthesized images, we adopt a three-layer discriminator, which effectively increase the receptive field. $\uparrow, \downarrow, +$ denote upsampling, downsampling, and add operation respectively.

to preserve global gland structures, such as shape and arrangement, while analyzing the finer details and textural information of the glands themselves, such as nuclei arrangement and lumen size. To achieve this, we propose to use a pyramid of conditional patch-GANs (Markovian discriminator) [54]. While similar pyramid architectures have been explored for natural image generation [61, 116, 136], we are the first to explore it on high-res, large-scale histopathology image synthesis and use it to augment data for segmentation.

5.2.1.1 Pyramid Generation

Our framework consists of a pyramid of conditional generators, $\{G_0, G_1, ..., G_N\}$. It is trained on a pyramid of image-mask pairs (x, y): $\{(x_0, y_0), ..., (x_N, y_N)\}$ where (x_n, y_n) is a down-sampled version of the original, (x_0, y_0). Each generator G_n attempts to generate realistic images $\tilde{x}_n$ conditioned on y_n. Through adversarial training, G_n learns to deceive an associated discriminator D_n, which attempts to distinguish (x_n, y_n) from $(\tilde{x}_n, y_n)$.

The pyramid framework begins at the coarsest scale G_N and proceeds sequentially to the finest scale G_0. Each G_n has the same architecture, and noise is injected at every scale to increase the variability among generated images. By progressing to finer scales throughout the generation process, the generators capture feature information of decreasing size. To start, G_N takes in a semantic map y_N with spatial white Gaussian noise z_N and maps it to an image $\tilde{x}_N$. At finer scales, G_n accepts an upsampled version of the generated image from the previous level $\tilde{x}^{\uparrow}_{n+1}$. The up-sampling is done via bi-linear interpolation. Spatial noise z_n is injected during this process, *i.e.*,

$$\begin{aligned} \tilde{x}_n &= G_n(\tilde{x}^{\uparrow}_{n+1}, y_n, z_n) \\ &= \tilde{x}^{\uparrow}_{n+1} + \Phi_n(y_n, z_n + \tilde{x}^{\uparrow}_{n+1}). \end{aligned} \tag{5.1}$$

Each of the generators G_n at finer scales ($n < N$) performs residual learning and adds details that are not generated by the previous scales, while maintaining features learned in previous steps of the pyramid. By going up in the generation process, finer details such as nuclei arrangement and lumen size are added while the global gland structures are preserved. Figure 5.1(b) illustrates the details of G_n.

5.2.1.2 Multi-layer Discriminator

Our discriminators take in image-mask pairs as input and differentiate whether they are real (x_n, y_n) or synthesized $(\tilde{x}_n, y_n)$ by the generator. To differentiate large-scale high-res real and synthetic images, the discriminator requires a large receptive field to stabilize training and improve generation performance. In practice, we found that using a multi-layer discriminator ([135]) increases the training stability. Specifically, for each discriminator D_n, we downsample the real and synthetic image-mask pairs by factors of two and four to create a pyramid. Then the discriminators operate at each step of the pyramid to differentiate whether they are real or synthetic. The discriminators have identical architectures at each scale. Similar to the generators, their receptive fields get smaller at each finer scale. The discriminator at the coarsest view guides the generator to generate images that are globally, spatially consistent images, thereby preserving the gland structure based on semantic masks. The discriminator at the finest scale encourages the generator to produce finer details within this consistent structure. The multi-layer discriminator is illustrated in Figure 5.1(d).

5.2.1.3 Training of Generation

Our model is trained sequentially, from the coarsest scale to the finest scale. Once each scale is trained, it is kept fixed. Our training loss consists of four parts: adversarial loss, reconstruction loss, feature matching loss, and perceptual loss, *i.e.*,

$$\begin{aligned} \min_{G_n} \max_{D_n} \mathcal{L}_{adv}(G_n, D_n) + \alpha \mathcal{L}_{rec}(G_n) + \\ \beta \mathcal{L}_{feat}(G_n) + \gamma \mathcal{L}_{perc}(G_n). \end{aligned} \tag{5.2}$$

Adversarial loss. The adversarial loss $\mathcal{L}_{adv}$ penalizes for the distance between the distribution of patches in (x_n, y_n) and the distribution of patches in generated sample $(\tilde{x}_n, y_n)$ through a Markovian discriminator.

Reconstruction loss. The reconstruction loss $\mathcal{L}_{rec}$ insures that the generator is able to

generate the original images based on the semantic mask.

$$\mathcal{L}_{rec}(G_n) = \left\| G_n(\tilde{x}_{n+1}^{\uparrow,rec}, y_n, z_n) - x_n \right\|^2. \tag{5.3}$$

Feature matching loss. The feature matching loss $\mathcal{L}_{feat}$ is incorporated to improve the stability of training. We extract features from multiple layers of D_n and learn to match these intermediate representations from (x_n, y_n) and $(\tilde{x}_n, y_n)$ by L_1 loss.

$$\mathcal{L}_{feat}(G_n) = E_{(x_n,y_n)} \left\| D_n^{(i)}(x_n, y_n) - D_n^{(i)}(\tilde{x}_n, y_n) \right\|_1. \tag{5.4}$$

Perceptual loss. The perceptual loss is also incorporated to ease the optimization [135]. We adopt a pre-trained VGGNet [119] for perceptual loss. Specifically, both the real and synthesized image-mask pairs are fed into a pre-trained VGGNet. We penalize the L_1 distance using features from the intermediate layers.

Note that $\mathcal{L}_{rec}$, $\mathcal{L}_{feat}$, and $\mathcal{L}_{perc}$ are only functions of G_n, *i.e.* we only use these losses to update G_n while keeping D_n fixed. We summarize the training algorithm in Algorithm 1.

Algorithm 1 Pyramid Generation

for Each scale of generation **do**

  for Number of training epochs **do**

    (1) Sample a batch of pairs $(\tilde{x}_n, y_n) \sim p_{G_n}(x_n, y_n)$ of size m_g, a batch of pairs $(x_n, y_n) \sim p(x_n, y_n)$ of size m_l;

    (2) Update D_n by ascending along its stochastic gradient based on Equation (5.2);

    (3) Update G_n by descending along its stochastic gradient based on Equation (5.2);

  end for

end for

5.2.2 Segmentation

To make the synthetic images useful for segmentation, we further design a pyramid structure with three players at each scale: (1) a segmenter S_n that characterizes the conditional distribution $p_{S_n}(\tilde{y}_n|x_n, \tilde{y}_{n+1}^{\uparrow}) \approx p(y_n|x_n)$, *i.e.* segmenting the input image based on the input image x_n and the semantic mask upsampled from the coarser level $\tilde{y}_{n+1}^{\uparrow}$; (2) a generator

G_n that characterizes the conditional distribution in the other direction $p_{G_n}(\tilde{x}_n|y_n, \tilde{x}^{\uparrow}_{n+1}) \approx p(x_n|y_n)$, *i.e.* generating image based on the mask y_n and the upsampled synthetic image from the coarser level $\tilde{x}^{\uparrow}_{n+1}$; and (3) a discriminator D_n that distinguishes whether a pair of image-mask comes from the true distribution $p(x_n, y_n)$. While G_n and D_n are parameterized the same way as in Section 5.2.1, we use a mini FC-DenseNet (m-FCDenseNet) ([58]) with 42 layers as S_n. The detailed architectures for G_n, S_n and D_n are shown in Appendix D.1. Figure 5.1(c) illustrates the schematic of S_n. As mentioned above, S_n is symmetric with G_n since it takes in a upsampled version of lower-scale segment results $\tilde{y}^{\uparrow}_{n+1}$ with the image x_n.

$$\tilde{y}_n = S_n(x_n, \tilde{y}^{\uparrow}_{n+1}, z_n). \tag{5.5}$$

Accordingly, D_n, as an adversarial part of S_n, takes pseudo image-mask pair from segmenter $(x_n, \tilde{y}_n)$ and distinguishes it from the real distribution (x_n, y_n). The adversarial component between S_n and D_n can be formulated as a minimax game:

$$\begin{aligned} &\min_{S_n} \max_{D_n} \mathcal{L}_{adv}(S_n, D_n) \\ \mathcal{L}_{adv}(S_n, D_n) &= E_{(x_n,y_n)}[\log(D_n(x_n, y_n)) \\ &\quad + E_{(x_{S_n},y_{S_n})}[\log(1 - D(S_n(y_n|x_n, \tilde{y}^{\uparrow}_{n+1})))]. \end{aligned} \tag{5.6}$$

However, the game defined in Equation (5.6) cannot guarantee that $p(x_n, y_n) = p(x_{s_n}, y_{s_n}) = p(x_{g_n}, y_{g_n})$ is the unique global optimum. To address this problem, we introduce the standard supervised loss for segmentation (*i.e.*, cross-entropy loss) in S_n, $\mathcal{L}_{ce}(S_n) = E_{(x_n,y_n)}[\log(p_{S_n}(y_n|x_n, \tilde{y}^{\uparrow}_{n+1}))]$. Consequently, the minimax game between S_n and D_n becomes:

$$\mathcal{L}_{ce}(S_n) = E_{(x_n,y_n)}[\log(p_{S_n}(y_n|x_n, \tilde{y}^{\uparrow}_{n+1}))]. \tag{5.7}$$

Consequently, the minimax game between S_n and D_n becomes:

$$\min_{S_n} \max_{D_n} \mathcal{L}_{adv}(S_n, D_n) + \mathcal{L}_{ce}(S_n). \tag{5.8}$$

5.2.2.1 Training of Segmentation

Training follows the same procedure as in Section 5.2.1.3, which starts from the coarsest scale and proceeds sequentially to the finest scale, except that for each scale we optimize D_n, G_n and S_n iteratively. Combining the minimax games defined in Equation (5.2) and Equation (5.8), we formulate the game with three players G_n, S_n, D_n as:

$$\begin{aligned}\min_{G_n,S_n}\max_{D_n}\mathcal{L}_{adv}(G_n,S_n,D_n)+\alpha\mathcal{L}_{rec}(G_n)+\beta\mathcal{L}_{feat}(G_n)+\\ \gamma\mathcal{L}_{perc}(G_n)+\alpha'\mathcal{L}_{ce}(S_n).\end{aligned} \tag{5.9}$$

The desired equilibrium of our model defined in Equation (5.9) is that the joint distributions defined by the segmenter S_n and the generator G_n at each scale both converge to the true data distribution [84]. This is an important property, as pointed out by Li *et al.*, because it ensures that the generator G_n generates realistic image-mask pairs, enabling the segmenter S_n to leverage the synthetic image-mask pairs for training. We provide the detailed theoretical analysis of the equilibrium in Appendix D.2.

It should be noted that, during the initial stage of training at each scale, the synthetic images from the generator G_n are not realistic enough for training the segmenter S_n due to their low quality. Therefore, these generated image-mask pairs are not used to train S_n until the number of epochs reaches a threshold such that G_n can generate reliable image-mask pairs. In practice, we hold the synthetic images for 100 epochs and then use them as normal labeled image-mask pairs for training S_n, except the coefficient for the cross-entropy loss is smaller compared to the real annotated data (see details in Section 5.2.4). The threshold is determined by visually inspecting the synthetic images. Using the synthetic image-mask pairs in early training stages can disrupt the optimization process and potentially hurt segmentation performance. We summarize the segmentation training algorithm in Algorithm 2.

Algorithm 2 Pyramid Generation and Segmentation

for Each scale of generation and segmentation **do**

for Number of training epochs **do**

(1) Sample a batch of pairs $(\tilde{x}_n, y_n) \sim p_{G_n}(x_n, y_n)$ of size m_g, a batch of pairs $(x_n, \tilde{y}_n) \sim p_{S_n}(x_n, y_n)$ of size m_s, and a batch of pairs $(x_n, y_n) \sim p(x_n, y_n)$ of size m_l;

(2) Update D_n by ascending along its stochastic gradient based on Equation (5.6);

(3) Update G_n by descending along its stochastic gradient based on Equation (5.6);

(4) Update S_n by descending along its stochastic gradient based on Equation (5.6);

end for

end for

Figure 5.1(a) illustrates our proposed pyramid model for histopathology image generation and segmentation. It has three advantages over the standard FC-DenseNet and U-Net, the current de facto model for medical image segmentation [111]. First, the synthetic image-mask pairs $(\tilde{x}_n, y_n)$ can be used as complementary data to train the segmenter S_n. Second, the discriminator D_n will enforce the S_n to generate good masks by matching segmented outputs and the annotated ground truth. Finally, by using the pyramid structure and being conditioned on the lower-level segmentation results, S_n increases its receptive field effectively. As a result, our model can be applied to high-res large-scale images without breaking the gland structures into different parts.

5.2.3 Semi-Supervised Segmentation

We further extend our framework to a semi-supervised learning (SSL) scenario. In SSL, we have a relatively small labeled set $(x_l, y_l) \sim p_l(x, y)$, and a large unlabeled set $x_u \sim p_u(x)$. We want to take advantage of the unlabeled data points x_u to boost our model's performance. To achieve this goal, we use the same architecture as in Section 5.2.2 with the following modification: at each scale, S_n takes in synthetic data, labeled data, and unlabeled data for training. We anticipate S_n to segment labeled data and synthetic data based on their masks (normal cross-entropy loss). For unlabeled images, we anticipate S_n to generate masks that are realistic enough that the image-mask pairs can confuse D_n through adversarial loss. The discriminator D_n accepts the image-mask pairs from the segmenter

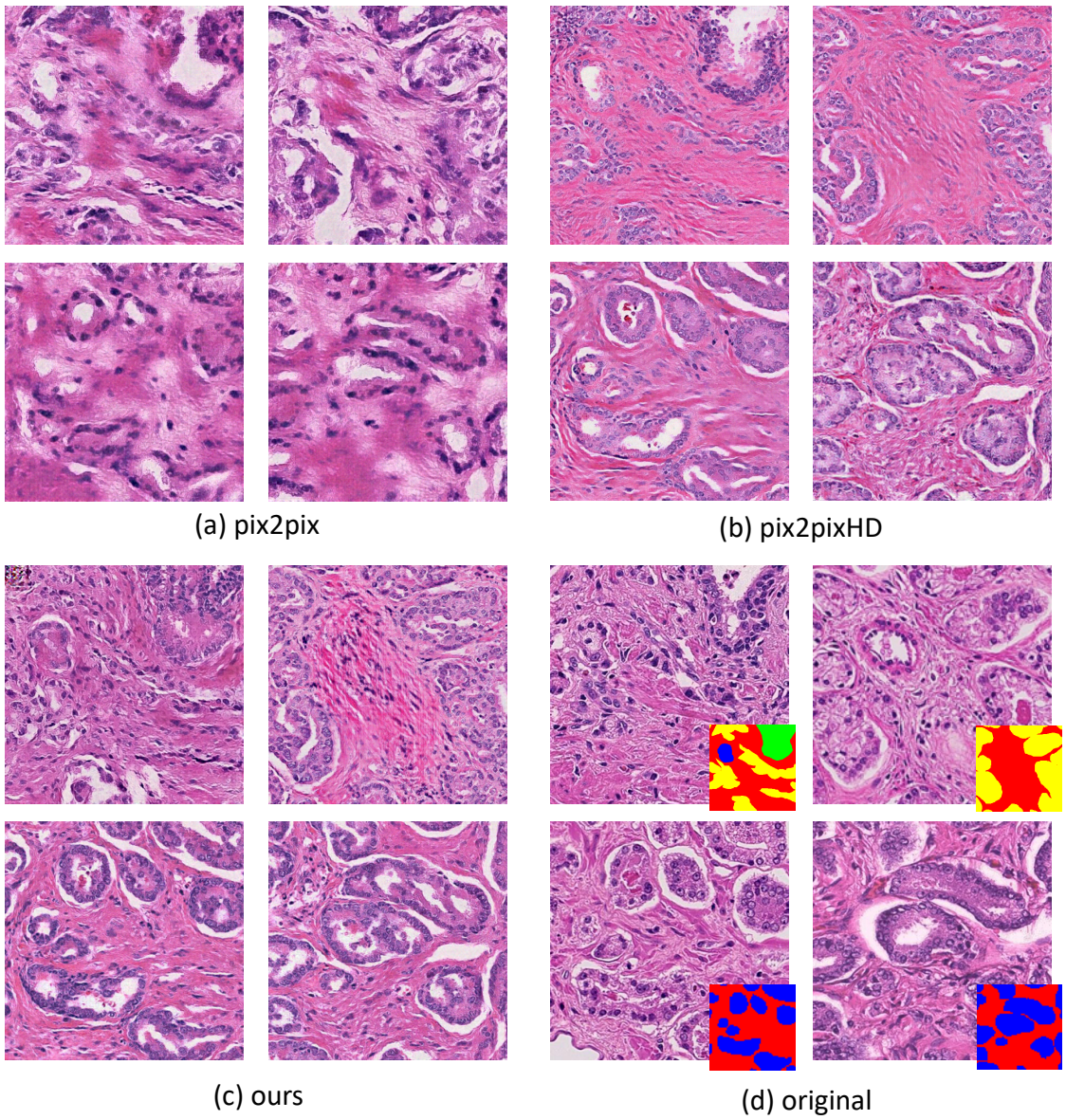

Figure 5.2: Randomly generated images by different models and the original real images. The figure illustrates that our model preserves the global structures of the semantic masks and generates sharper images with finer details than the baselines.

$S_n(x_{S_n}, y_{S_n}) \sim p(x_{u,n})p_{S_n}(y_{u,n}|x_{u,n})$, the generator $G_n(x_{G_n}, y_{G_n}) \sim p(y_n)p_{G_n}(x_n|y_n)$, and from the labeled data distribution $(x_{l,n}, y_{l,n}) \sim p_l(x_n, y_n)$ for judgement. D_n treats the labeled data as positive samples, and the pairs from both G_n and S_n as negative samples. By doing so, G_n and D_n, and S_n and D_n form two sets of adversarial training. The discriminator D_n

will enforce both $S_n(x_{S_n}, y_{S_n}) \sim p(x_{u,n})p_{S_n}(y_{u,n}|x_{u,n})$ and $G_n(x_{G_n}, y_{G_n}) \sim p(y_n)p_{G_n}(x_n|y_n)$ to match with $(x_{l,n}, y_{l,n}) \sim p_l(x_n, y_n)$ during the training process, thus we will have a good generator G_n and good segmenter S_n at the end of the training.

One key problem in SSL is the limited amount of labeled data. A powerful D_n may memorize the labeled data and reject other types of samples. Consequently, G_n may collapse to these modes. To address this problem, we adopt the practical techniques introduced in ([16, 84]). We generate pseudo masks through S_n for some unlabeled data and randomly choose these pairs as positive samples of D_n. This process introduces some bias to the target distribution of D_n, but it gives D_n a better chance to model the complete data distribution([16, 84]). Moreover, since S_n converges much faster compared to G_n, this operation enables G_n to explore a much larger image-mask distribution that includes both the labeled and unlabeled data information. In other words, S_n is able to provide pseudo masks for the unlabeled image x_u, while D_n will judge if the pseudo masks are reliable or not. This in turn will affect the evolution of G_n, which will take advantage of the unlabeled image to generate high quality images-mask pairs. These synthetic image-mask pairs that implicitly contain unlabeled data information will eventually benefit the training of the segmenter S_n. We will demonstrate that it serves as a key for performance improvement in SSL. We summarize the entire training procedure for SSL in Algorithm 3.

Algorithm 3 Pyramid Semi-supervised Segmentation

for Each scale of generation **do**

for Number of training epochs **do**

(1) Sample a batch of pairs $(\tilde{x}_g, y_g) \sim p_{G_n}(x_n, y_n)$ of size m_g, a batch of labeled pairs $(x_l, y_l) \sim p(x_l, y_l)$ of size m_l, and a batch of unlabeled pairs $x_u \sim p(x_u)$ of size m_u;

(2) Input x_u to S_n and get $(x_u, y_u) \sim p_{S_n}(x_n, y_n)$, input (x_l, y_l) to S_n and get $(x_l, \tilde{y}_l) \sim p_{S_n}(x_n, y_n)$;

(3) Input $(x_l, y_l) \sim p(x_l, y_l)$, $(x_u, y_u) \sim p_{S_n}(x_n, y_n)$, $(x_l, \tilde{y}_l) \sim p_{S_n}(x_n, y_n)$, and $(\tilde{x}_g, y_g) \sim p_{G_n}(x_n, y_n)$ to D_n to get the output;

(4) Update D_n by ascending along its stochastic gradient based on Equation (5.9);

(5) Update G_n by descending along its stochastic gradient based on Equation (5.9);

(6) Update S_n by descending along its stochastic gradient based on Equation (5.9);

end for

end for

5.2.4 Implementation Details

We train the proposed model on a single Tesla V100S GPU with 32GB memory. We set the learning rates for G_n, S_n, and D_n to 1×10^{-4}, 5×10^{-4}, and 5×10^{-4}, respectively. We use an Adam optimizer and employ $\beta_1 = 0.5$ and $\beta_2 = 0.999$. We set the hyperparameters $\alpha, \beta, \gamma, \alpha'$ in Equation (5.9) to be $0.001, 10, 10, 1$, respectively, same as in pix2pixHD and SinGAN [116, 135]. We do not further tune these hyperparameters, as they provide good generation and segmentation results. Once the synthetic data are used to train S_n, we set the coefficient of cross-entropy loss to 0.03 in order to decrease the adversarial effect of imperfect synthetic data. To choose the batchsize m_l, m_g, m_u in Algorithm 3, we follow the principle discovered by Li *et al.* [84] and set $m_l = 8, m_g = 4, m_u = 10$.

5.3 Experiments

In this section, we will first introduce the two datasets we used in our experiments, the GalS and Prostate Gleason grading datasets. Then, we evaluate our proposed method in three aspects: image synthesis, image segmentation, and semi-supervised segmentation. We compare

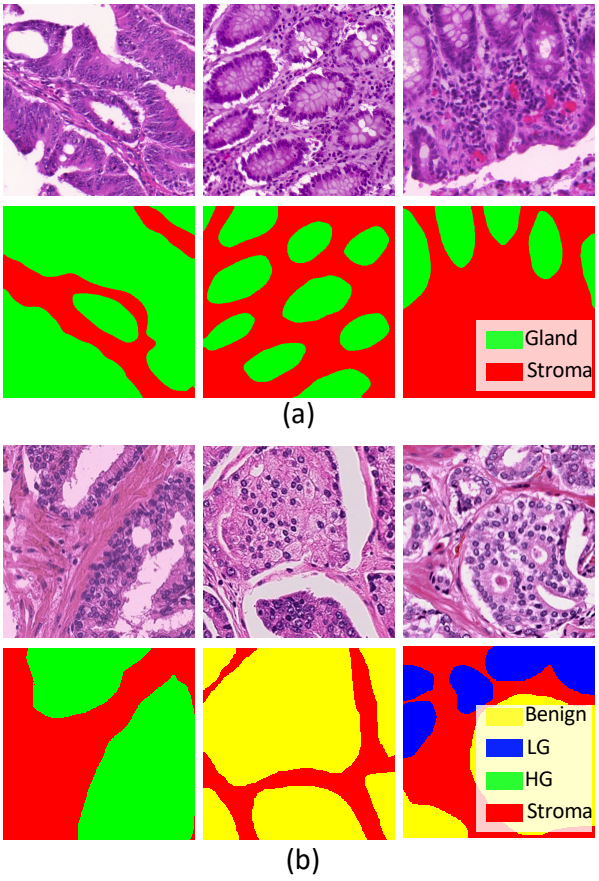

Figure 5.3: Samples from the GalS and Prostate datasets. Three representative examples are shown from each dataset. (a) Samples from GalS dataset with their segmentation ground truth. Green color indicates the gland in the images while red color indicates stroma. (b) Samples from Prostate dataset with their segmentation ground truth. Images are annotated by pathologists for stroma in red, benign glands in yellow, low-grade cancer in blue, and high-grade cancer in green.

our method with baseline models, including pix2pix, pix2pixHD in image generation, mini FCDenseNet (m-FCDenseNet), U-Net, DCAN *etc.* in segmentation, and demonstrate the effectiveness of our method under different scenarios (fully-supervised v.s semi-supervised).

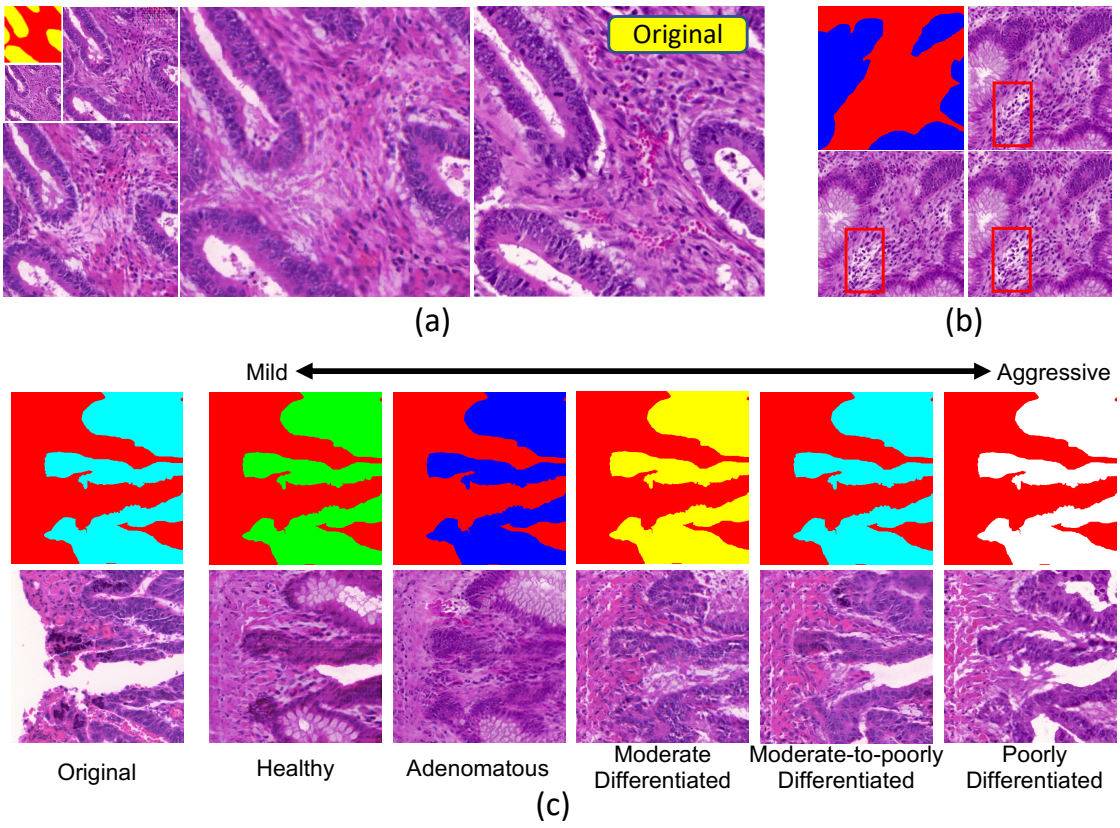

Figure 5.4: (a) Generated coarse-to-fine results trained on the GlaS dataset. (b) Three generated images based on the same mask. Noise is injected during generation so that the model can synthesize images with variations. Clearer variations can been seen in the video clip in SI. (c) Image manipulation on synthesized images. Different gland types are observed when we changed the label from healthy to poorly differentiated.

5.3.1 Datasets

Our experiments are conducted on two histopathology image segmentation datasets, including the GlaS dataset [120] and the prostate Gleason Grading dataset [30, 52].

5.3.1.1 GlaS Dataset

The GlaS dataset [120] was acquired by a team of pathologists at the University Hospitals Coventry and Warwickshire, UK. It consists of a training set with 85 images and a testing

set with 80 images for colorectal cancer. The majority images are 775 × 522 pixel patches from whole-slide histology images of the colon. These images are scanned by Zeiss MIRAX MIDI and set to 20X magnification. The output was a color RGB image with the pixel size of 0.62 μm × 0.62 μm. Along with the images, pixel-wise annotations for epithelial glands (binary masks) and a spreadsheet detailing the type of the glands are provided. The types of glands are characterized as healthy, adenomatous, moderately differentiated, moderate-to-poorly differentiated, and poorly differentiated. The test dataset is divided into two subsets; subset A (60 images) released earlier and subset B (20 images) released during the original MICCAI workshop in 2015. We report results on the combined test set and the individual subsets.

5.3.1.2 Prostate Gleason Grading Dataset

The prostate Gleason grading dataset [30, 52] consists of 513 images. The dataset is retrieved from archives in the Pathology Department at Cedars-Sinai Medical Center (IRB# Pro00029960). The 513 images are combined from two sets of tiles. 224 of the images are from 20 patients and contain stroma (ST), benign or normal glands (BN, rated as GG2 or below), low-grade cancer (LG, image areas rated as GG3) and high-grade cancer (HG, image areas rated as GG4) (subset A). The remaining 289 images are from 20 different patients and contain dense high-grade tumors including Gleason grade 5 (GG5) as well as Gleason grade 4 (GG4) with cribriform and non-cribriform glands (subset B). Slides from subset A were digitized using a high resolution whole slide scanner SCN400F (Leica Biosystems, Buffalo Grove, IL), whereas slides from the subset B were acquired through the Aperio scanning system (Aperio ePathology Solutions, Vista, CA). The scanning objective in both systems was set to 20x. The output was a color RGB image with the pixel size of 0.5 μm×0.5 μm and 8 bit intensity depth for each color channel. Representative tiles were extracted from whole slide images as 1200 × 1200 pixel tiles for analysis. The content of each tile was hand-annotated by an expert research pathologist using an in house developed graphical user interface. We use 80% of the images as training with the remaining 20% as testing unless

otherwise specified.

5.3.1.3 Pre-processing

To handle different image sizes, we tiled the images into squares with overlap but without any scaling. Specifically, we tiled the GlaS images to 512×512, resulting around 500 patches, the prostate images to 1024×1024, resulting around 1000 patches. The intensity value of the images was normalized to $[-1, 1]$. At training time we applied flipping, rotation, and color jitter to augment the data. When scaling, bi-linear interpolation was used for images while nearest neighbor method was used for masks. Figure 5.3 shows some representative images of the cropped patches from both datasets.

5.3.2 Image Generation

5.3.2.1 Qualitative Evaluation

We first show the generation results of our model qualitatively. For the GlaS dataset, the model was trained on 512×512 image patches on two types of masks: binary masks (stroma *vs*. epithelial glands) and multi-category masks that indicated the gland type. Here, we made a minor assumption that all the glands in one single patch have the same type as indicated in the data spreadsheet. When training the model, we first downsampled the original images and started from patches of size 64×64. For each following scale, we multiplied the image length by a factor of two. Thus it led to a four-level training scheme with image size of $64^2, 128^2, 256^2, 512^2$. For the prostate Gleason grading dataset, the model was trained on 1024×1024 image patches starting from 64×64, which led to a five-level training scheme of $64^2, 128^2, 256^2, 512^2, 1024^2$. We compared our method with two baseline methods: pix2pix generation [54] and pix2pixHD generation [135]. To qualitatively analyze the results, we show samples of synthetic images in Figure 5.2. The figure illustrates that our method preserves the global structure indicated by the semantic mask, and generates sharper images with finer details than the baseline methods. More synthetic high-resolution samples can be

found in Appendix D.3.

Next, we demonstrate some interesting aspects during the model's generation process. Figure 5.4(a-c) shows the generation results from the GlaS dataset conditioned on multi-category masks. Figure 5.4(a) shows the coarsest-to-finest generation process. We observe that as the training process progresses, the generated images are honed (*i.e.* more details are added while the gland structures are preserved) so that the images look more realistic. Figure 5.4(b) shows three images that are generated based upon the same mask. Once training was completed, we performed inference using the mask shown in the top-left corner. The generation process also started from 64×64 patch size and then went up to 512×512. Though all three of the images preserve global structures, subtle details are different due to the injected noise, *e.g.* the stroma details in the rectangle in Figure 5.4(b). It can be more easily observed in the animation provided in the Appendix, where we cycle through these generated images. Since we injected noise during the generation process, we can continually generate images based upon the same mask and use them for training in the segmentation task. Figure 5.4(c) shows image manipulation results by changing the mask from healthy to poorly differentiated. For these images, we changed the gland labels on the input masks and fed them into our generation framework to generate images of different grades with the same gland boundaries. It suggests that the generator can learn meaningful latent representations instead of simply memorizing the training data. Similar observations can be found for the Prostate dataset, where we changed the labels from low grade to high grade and vice versa. For more results on prostate dataset generation, noise injection, and image manipulation, we refer readers to Appendix D.3.

5.3.2.2 Quantitative Evaluation

If our generated images are realistic looking, then their distribution should be indistinguishable from that of the real images. Therefore, we can quantitatively evaluate the quality of the synthetic images by computing the Frechet Inception Distance (FID) between the distributions of real images and synthetic images ([46]). Lower values of FID indicate

Table 5.1: FID score for image generation.

	GalS	Prostate
pix2pix	3.654	4.354
pix2pixHD	1.028	2.577
Ours	0.0059	0.013
Original images	8.4×10^{-4}	2.3×10^{-4}

the distribution are more similar, implying more realistic-looking images. Specifically, we adopted a ResNext50 model pre-trained on large-scale histopathology images to extract features for computing FID. We didn't use the Inception v3 model that commonly used in other literature, as we could not find an off-the-shelf Inception model pre-trained on histopathology images. Therefore, our FID values are not directly comparable to common FID values in other studies. We provide the FID score of our model and other baselines in Table 5.1. As shown in Table 5.1, our model achieves lower FID compared to the baselines in both dataset. To make the comparison more meaningful, we also provide the FID values for the original images as a comparison. We calculated the FID of original images by randomly dividing the images into two groups. The score represents a level of best generation performance possible measured by FID. The FID quantities imply that our model generates more realistic images compared with the baseline models. More details regarding on how we calculate FID score can be found in Appendix D.4.

5.3.3 Segmentation

We examine whether our proposed method can boost the performance in a fully-supervised segmentation task and reveal the contribution of each component to the performance through an ablation study.

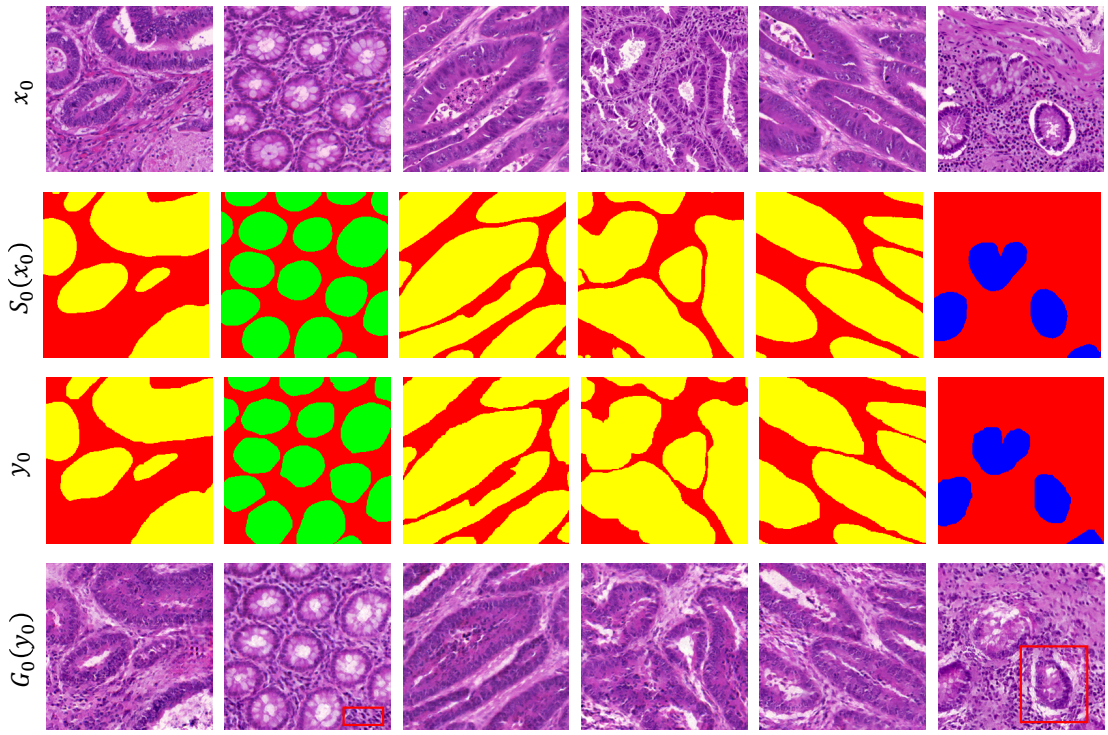

Figure 5.5: Segmentation and generation results under fully-supervised scenario. x_0 are the original images; $S_0(x_0)$ are the semantic segmentation results by S_0; y_0 are the ground truth segmentation annotation; $G_0(y_0)$ are the synthetic images by G_0 conditioned on y_0.

5.3.3.1 Fully-supervised Segmentation Results

We first implemented the segmentation model as discussed in Section 5.2.2 in fully-supervised fashion. In supervised learning, we used the full training dataset, while using image synthesis to augment the training sets. Table 5.2 shows the segmentation performance of our model on the GalS dataset compared with other studies. The performance shown in the table is based on the binary ((stroma *vs*. epithelial glands)) segmentation task, which is the same as in MICCAI Challenge 2015. We report the GlaS challenge metrics [120] including object-level F1 score, object-level dice coefficient, and object-level Hausdorff distance, and compare them with other main studies in literature. As shown in Table 5.2, we achieved the second best performance among all other studies, though our method does not surpass the state-of-the-art performance ([39]). We also present the segmentation results in Figure 5.5 (row 1-3).

We want to point out that instead of cropping the images to small patches, stitching back after the segment inference, and performing tedious post-processing as all other studies did, our method directly applies the segmentation model on the original 512×512 size. A direct comparison between our proposed method and m-FCDenseNet (backbone of S_n) reveals our model is able to boost the performance by a large margin when applied on high-res large-scale images directly. Similar findings are observed for the Prostate dataset, which we list in Appendix D.3.

5.3.3.2 Ablation Study

As we mentioned above, our proposed fully-supervised segmentation model is different from the traditional segmentation method in two ways. First, our method consists of a pyramid structure for generation and segmentation. Therefore, we can perform segmentation on large histopathology images and do not need to tile the image and stitch them back together. Thanks to the pyramid structure, the final segmentation network S_0 has a larger receptive field that makes it able to consider both the large gland structures and finer nuclear details simultaneously. By skipping the tiling process, we are also able to avoid splitting the gland structure into different tiles and deteriorating the prediction accuracy. Second, our method leverages the synthetic images from G_n as augmented data to train S_n. It enlarges the training set size and is expected to improve the segmentation performance.

To determine how these two aspects affect the final performance, we perform an ablation study. As shown in Table 5.2 of column "100%", the first three rows are all operating on 512×512 images. In first row *m-FCDenseNet*, we only have a single level S_0 applied to 512×512 image. Due to the limited receptive field, *m-FCDenseNet* alone has the worst segmentation performance. The *m-FCDenseNet+pyramid* model (row two) has the same pyramid structure as our model except that the synthetic images are not used to train S_n. Compared with *m-FCDenseNet* alone, the pyramid structure can effectively enlarge the receptive filed, leading to a performance improvement by a large margin. Conversely, comparing *m-FCDenseNet+pyramid* (row two) and our full model (row three), we observe

the improvement is marginal, which indicates the synthetic images are not the key for performance boost in fully-supervised settings. It explains why our model does not surpass the state-of-the-art results. As our large receptive field, led by the pyramid structure, can be achieved by tiling the image to small sizes equivalently. The performance improvement of our model is negligible in fully-supervised settings. We will further discuss this issue in Section 5.4.3.

Table 5.2: GlaS challenge metrics for the total test set and subsets (A, B). * denotes methods that are operating on 512 × 512 scale.

Method	Object Dice (A, B)	F1 Score (A, B)	Hausdorff (A, B)
m-FCDenseNet*	0.748 (0.731, 0.792)	0.676 (0.662, 0.710)	123.39 (122.4, 125.9)
m-FCDenseNet + pyramid *	0.870 (0.894, 0.822)	0.860 (0.878, 0.776)	65.7 (54.1, 108.3)
Ours*	**0.874 (0.895, 0.825)**	**0.866 (0.890, 0.803)**	**61.85 (50.3, 100.4)**
FCN-8 [89]	0.781 (0.795, 0.767)	0.763 (0.783, 0.692)	124.2 (105.0, 147.3)
DeepLab [14]	0.833 (0.859, 0.804)	0.813 (0.862, 0.764)	96.2 (65.7, 124.9)
Seg-Net [4]	0.838 (0.864, 0.807)	0.806 (0.858, 0.753)	92.6 (62.6, 118.5)
U-Net [111]	0.868 (0.884, 0.819)	0.841 (0.865, 0.768)	69.6 (55.6, 111)
DCAN [13]	0.868 (0.897, 0.781)	0.863 (0.912, 0.716)	74.2 (45.4, 160.3)
Graham [39]	**0.902 (0.919, 0.849)**	**0.896 (0.920, 0.824)**	**54.7 (41.0, 95.7)**

5.3.4 SSL-Segmentation

In this section, we examine whether our proposed method can boost the performance in a semi-supervised segmentation task and analyze how the pyramid structure and synthetic data augmentation contribute to the final performance.

5.3.4.1 SSL-Segmentation Results

We implemented SSL-segmentation and evaluated it by varying the amount of labeled data provided for training as commonly used by the literature [16, 84]. The labeled data used for training were randomly selected. In our experiments we used 20%, 40%, 60%, 80% and the

Table 5.3: mIOU for SSL-segmentation on GalS dataset. All the results are generated under inductive learning unless specified in the parenthesis.

Task	Method	20%	40%	60%	80%	100%
Binary	m-FCDenseNet	0.674	0.686	0.701	0.750	0.774
	m-FCDenseNet+pyramid	0.717	0.730	0.767	0.806	0.810
	Ours	0.793	0.799	0.817	0.823	0.827
	Ours (transductive)	**0.817**	**0.824**	**0.845**	**0.830**	**0.849**
Multi-Category	m-FCDenseNet	0.203	0.254	0.262	0.282	0.290
	m-FCDenseNet+pyramid	0.216	0.259	0.289	0.325	0.358
	Ours	0.242	0.287	0.301	0.336	0.368
	Ours (transductive)	**0.325**	**0.341**	**0.386**	**0.375**	**0.383**

full training dataset as labeled data and the rest as unlabeled data to train the model. For fair comparison, we also used m-FCDenseNet as a baseline. Furthermore, we conducted both inductive learning and transductive learning for our models, where in transductive learning the images in the test set were also treated as unlabeled data points for training. For GalS dataset, we performed two sets of SSL experiments: one for a binary segmentation task, where we only used a binary mask (stroma v.s epithelial gland) for training; and the other for a six-category segmentation task, where the mask not only contains the information of gland location but also the type of gland (healthy, adenomatous, moderately differentiated, moderate-to-poorly differentiated, or poorly differentiated). We used mean intersection over union (mIOU) as a metric to evaluate the segmentation performance in both experiments. The results are presented in Table 5.3. Note that all the results are generated under inductive learning unless specified in parentheses. As can be seen, our model outperforms the m-FCDenseNet in all the cases of varying the amount of training data, demonstrating the effectiveness of our model. Transductive learning outperforms inductive learning as expected, since transductive learning incorporates the testing data as unlabeled data in training. More results of SSL-segmentation on Prostate dataset can be found in Appendix D.3.

5.3.4.2 Ablation Study

To determine the effectiveness of the pyramid structure and synthetic data augmentation in the SSL setting, we present the binary segmentation results in column chart as shown in Figure 5.6. In this figure, we only focus on inductive learning cases for fair comparison. Specifically, we calculate the performance gap between the single m-FCDenseNet baseline and our model and define two δ's as δ_1 to be *normalized performance improvement between* m-FCDenseNet *and* m-FCDenseNet + pyramid, and δ_2 to be *normalized performance improvement between* m-FCDenseNet + pyramid *and* our method. Intuitively, δ_1 roughly characterizes the contributions from the pyramid structure and δ_2 roughly characterizes the contribution from synthetic data augmentation. As can be observed, δ_1 gradually increases as we increase the amount of labeled data, while δ_2 gradually decreases. This result indicates that in the low-labeled data scenario, synthetic data augmentation plays a more important role than the pyramid structure. As we have more labeled data, the pyramid structure becomes the key factor of performance improvement compared with m-FCDenseNet. Similar trends have been observed in other SSL-experiments (see Appendix D.3). We will further discuss the effectiveness of synthetic data augmentation in Section 5.4.3.

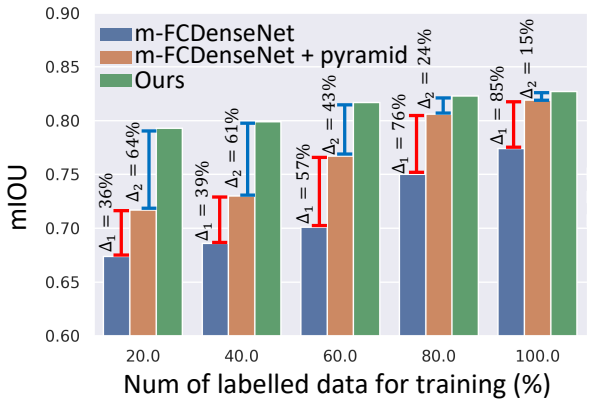

Figure 5.6: Analysis of SSL-segmentation results on GalS dataset. The experiments are done by using 512×512 images for binary segmentation task.

5.4 Discussion

5.4.1 Image Generation

As shown both quantitatively (Table 5.1) and qualitatively (Figure 5.2), our model generates more realistic images compared to the baseline models. The pix2pix method only leverages traditional conditional generative adversarial networks with a single Markovian discriminator (PatchGAN). It is more suitable for small-size images synthesis since the receptive field for both the generator and discriminator is limited. As a result, the generation of local patches is unaware of the global structure, potentially leading to spatial inconsistency. Pix2pixHD adopts a multi-scale generator and discriminator to capture the image features of different scales. As a result, it generates more realistic images with higher spatial consistency (*e.g.* gland structures are distinguishable from stroma). In contrast, our proposed model provides a pyramid structure, such that different scales focus on generating features of different levels. By conditioning on the generated images of the previous scale, our model is able to add finer details to the generated images while preserving the gland structure based on the semantic masks. As a result, our synthetic images are better in spatial consistency (compared with pix2pix model), and sharper with finer details (compared with pix2pixHD model).

5.4.2 Image Scales for Segmentation

Input image patch size is a key factor for segmentation performance. Therefore, to achieve good performance in histopathology image segmentation, researchers often have to tile the large-scale histopathology images into small patches and design a network structure with suitable receptive field. As a result of tiling, gland structures are often split into different parts, which deteriorates the segmentation accuracy. One of the merits of our proposed method for segmentation is that our model is not as sensitive to the input image scales compared to the single model with fixed receptive field. To demonstrate this, we evaluated our model from 64×64, up to 512×512 images on the GalS dataset for the binary segmentation task. We compared it with m-FCDenseNet, which has the same architecture at a single scale

S_n. Figure 5.7 illustrates the results. In general, as we increase the image size, we expect the segmentation accuracy to increase because the resolution becomes higher. As can be seen, our model performs similar as m-FCDenseNet on small scale images. Once we input 512×512 images, the performance of our model increases while m-FCdenseNet suffers a sharp drop (see the red rectangle in Figure 5.7). Since our model is relatively insensitive to the input image size, it is able to process large histopathology images with high magnification without breaking the gland structures into different tiles, which has been demonstrated to improve the segmentation accuracy in both fully-supervised and semi-supervised scenarios (see ablation study in Section 5.3.3 and Section 5.3.4).

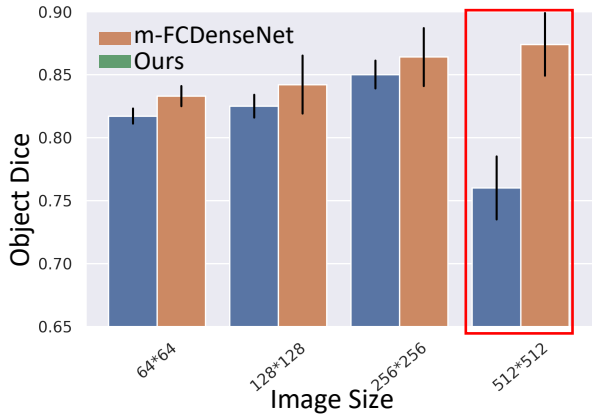

Figure 5.7: Model performance with different input image scales. Our model is less sensitive to image scale compared to single-scale model such as m-FCDenseNet.

5.4.3 Effectiveness of Synthetic Data

As briefly discussed in Section 5.3.3 and Section 5.3.4, we found that the synthetic data augmentation is not always helpful. In general, synthetic data augmentation is more effective in SSL, especially when the labeled images are extremely limited. These observations are aligned with other studies ([36]), where people find that generating a lot of additional samples by GAN and use them to provide a bigger dataset to train a classifier does not improve performance. The reason behind it is that it requires the generator to generalize

better than the classifier, which is hard to achieve if you train both the generator and classifier on the same dataset. In this case, the generator gets no extra information compared to the classifier. It explains why our proposed model with synthetic data augmentation does not provide significant improvement compared to the *m-FCDesnseNet + pyramid* baseline in supervised settings. In fact, we argue that the normal data augmentation methods, such as flipping, rotation, color jitters *etc.*, are enough to provide strong regularization. Though the synthetic data augmentation provide extra regularization, the performance improvement is almost negligible (see Table 5.2 and Table 5.3 in column "100%").

On the contrary, synthetic data augmentation works well in SSL as discussed in Section 5.3.4. Specifically, We generate pseudo masks through S_n for some unlabeled data and randomly choose these pairs as positive samples of D_n. This process introduces some bias to the target distribution of D_n, but it gives D_n a better chance to model the complete data distribution. In return, it enables G_n to explore a much larger image-mask manifold that includes both the labeled and unlabeled data information. In other words, G_n generated image-mask pairs are able to provide extra information gains compared with the labeled training sets. It also explains why we observe δ_2 is larger in the low-data scenario, while it is negligible when we use 100% labeled data for training. Intuitively, the larger information gains G_n can provider, the bigger improvement the synthetic data augmentation can contribute.

5.4.4 Limitations and Future Work

5.4.4.1 Exploring Meaningful Latent Representations

Besides the image generation results, we also provide image manipulation results by changing the gland labels in Figure 5.4(c). It suggests that the generator G_n is not memorizing the training data itself but learning useful representations that are predictive for clinically relevant measurements. Nevertheless, in this study we do not provide any quantitative analysis on the learned representations. In future work, we plan to make the latent representations

more explainable and associate them with clinically relevant measurements through mutual information maximization. We also plan to seek help from pathologists to provide clinically relevant measurements in future work.

5.4.4.2 Increasing Memory Efficiency

At each scale, our proposed model consists of three players, and because the algorithm has to maintain the weights of previous scales during training, the current model can occupy a lot of memory in GPU. Currently, when implementing the algorithm in a single Tesla V100S GPU, we are only able to generate 1024×1024 images using G_n and D_n alone. By incorporating S_n, we can only process image with sizes up to 512×512 at best. Therefore, potential future work is to increase the memory efficiency of the proposed method. It can be improved by two ways: first, we can take advantage of more memory-efficient network modules as backbone, such as EfficientNet *etc.*; second, we can develop a random selection process to make the finer scale only focus on a sub-volume of images.

5.4.4.3 Improving Segmentation Results

As we demonstrated, our model does not surpass the state-of-the-art results in the fully supervised case. However, our model is complementary to the current state-of-the-art methods. For instance, the rotated convolution kernels used by [39] can also be applied to S_n to increase performance. In the future, we would like to incorporate other ideas to improve the segmentation performance.

Additionally, we also found that there are generation artifacts in the synthetic images. For example, the synthetic image may miss a small part of a gland (see Figure 5.5 column 2 red rectangular area), or the synthetic gland may not have a clear boundary as the real gland does (see Figure 5.5 column 4 red rectangular area). These artifacts can potentially deteriorate the performance of segmentation, as we use the ground truth mask along with the synthetic images to train S_n. In the future, we would like to study the generation artifacts with the help of pathologist and improve the quality and diversity of synthetic images.

5.4.4.4 Investigating Active Learning

We performed multiple runs for SSL-training on the prostate dataset (see in Appendix D.3). For each run, we randomly selected a number of images as labeled data with the rest as unlabeled. We found that the variance of model performance increased as we decreased labelled training data, *i.e.* the performance variance was larger when we only used 20% training data compared to the whole training set. It indicates the importance of selected labeled data in the initial training stage. The results are not surprising and are related to active learning. In active learning, we have to develop a model to identify the most "important," "typical" images for expert to annotate. In this way, we can stabilize the performance in the low-labelled data regime. A potential future work could be extending our model for active learning.

5.5 Conclusion

In this work, we present a novel pyramid framework for synthesizing high-res histopathology images and use it to augment a dataset for a segmentation task in both supervised and semi-supervised scenarios. We provide detailed analysis on our synthetic images both qualitatively and quantitatively. We also demonstrate how the pyramid structure and synthetic data augmentation contribute to the final model performance differently. We conclude that GANs can be effectively used to augment small pathology datasets to improve semantic segmentation in semi-supervised settings, which could potentially enhance downstream clinical analysis. We anticipate our findings can shed the light to the future researches on low-cost, high-res, large-scale histopathology image analysis.

CHAPTER 6

PathAL: An Active Learning Framework for Histopathology Image Analysis

6.1 Introduction

In this chapter, we investigate an active learning framework, called PathAL, that is tailored to histopathology image analysis. To reduce the required number of expert annotations, PathAL selects two groups of unlabeled data in each training iteration: one "informative" sample that requires additional expert annotation, and one "confident predictive" sample that is automatically added to the training set using the model's pseudo-labels. To reduce the impact of the noisy-labeled samples in the training set, PathAL systematically identifies the noisy samples and excludes them to improve the generalization of the model. Our model advances the existing AL method for medical image analysis in two ways. First, we present a selection strategy to improve classification performance with fewer manual annotations. Unlike traditional methods focusing only on finding the most uncertain samples with low prediction confidence, we discover the large amount of high confidence samples from the unlabeled set and automatically add them for training with assigned pseudo-labels. Second, we design a method to distinguish between noisy samples and hard samples using a heuristic approach. We exclude the noisy samples while preserving the hard samples to improve model performance. Extensive experiments demonstrate that our proposed PathAL framework achieves promising results on a prostate cancer Gleason grading task, obtaining similar performance with 40% fewer annotations compared to the fully supervised learning scenario. An ablation study is provided to analyze the effectiveness of each component in PathAL,

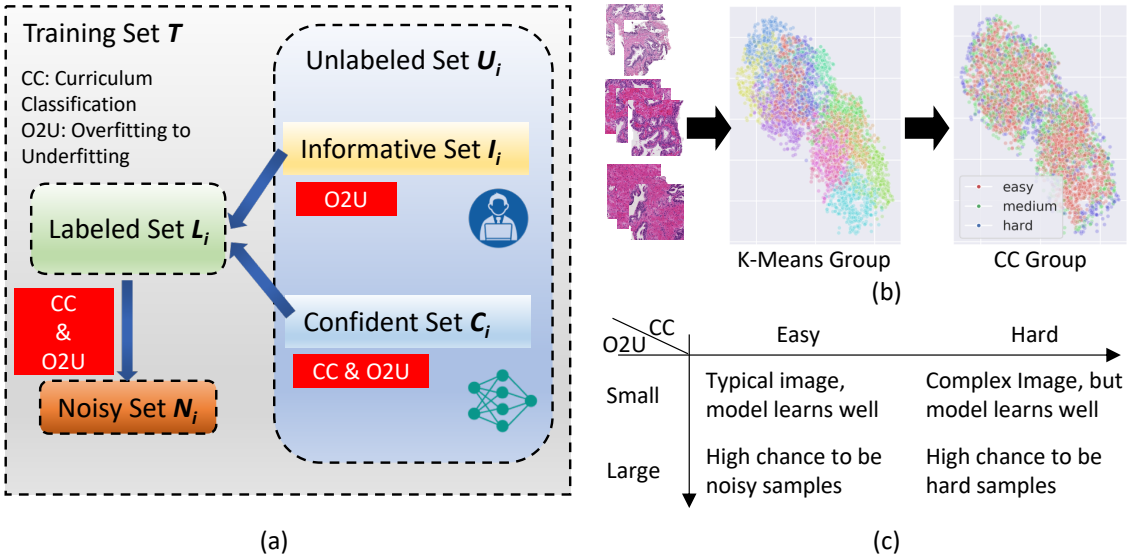

Figure 6.1: (a) Schematics of our proposed PathAL. The core algorithm of PathAL consists of three steps in the ith iteration: discarding noisy samples N_i, requesting human experts to annotate informative samples I_i and adding them to L_{i+1}, adding confident predictictive samples C_i with their "pseudo-labels" to L_{i+1}. The curriculum classification (CC) algorithm and overfitting to underfitting (O2U) monitor are used to select N_i, I_i, C_i. (b) Illustration of the CC algorithm. Tissues from one slide are mapping to one single point in deep feature space, where K-Means Clustering is used to group them in subsets. The CC algorithm is applied to each subsets and classify the image complexity to "easy", "medium" and "hard" based on their local density. (c) Principles on how to determine N_i and C_i based on CC and O2U results. A sample that is classified as "easy" based on its complexity but has large training loss variation is more likely to be annotated wrong; while if it is classified as "hard" for its complexity, it is more likely to be a hard sample. Conversely, if a sample is classified as "easy" on its complexity, and the variation of its predictive entropy is low by the current model, we will have a higher confidence that the current prediction is correct.

and a pathologist reader study is conducted to validate our proposed algorithm.

6.1.1 Motivation

Deep neural networks (DNNs) have achieved great success in a wide variety of medical image analysis tasks [88]. However, noise-free expert annotations are crucial to achieve high performance. Unfortunately, in medical image analysis, obtaining enough annotations can be expensive and time-consuming for many tasks. In histopathology images analysis, the size of the collected dataset can be large, but performing annotations requires years of professional training and domain knowledge. In addition, the labels provided by different pathologists can demonstrate low inter-reader variability. For example, in prostate cancer grading using Gleason scoring, the concordance rate of multiple pathologists can be as low as 57.9% [144], which results in noisy annotations. DNNs are capable of fitting to noisy annotations, but they may not generalize to unseen data, which is an important component of clinical applications. Furthermore, it is challenging to distinguish mislabeled samples from hard samples. Mislabeled samples are samples with incorrect annotations, while hard samples have the correct label, but the samples themselves are not "typical." The lack of large and noise-free annotation sets is a significant challenge in histopathology image analysis, preventing DNNs to scale to the size of collected data.

Recent studies have investigated methods for dealing with annotation challenges in medical imaging. One solution is to use active learning (AL) [10]. AL aims to reduce the amount of labeled data necessary for the learning task. It employs various sampling methods to select samples from an unlabeled set. The selected samples are then annotated by experts and used to train the model. A carefully designed sampling method can reduce the overall number of labeled data points required to train the model and make the model robust to class imbalances. However, traditional AL methods do not address the noisy label issue.

A few studies have also sought to detect noisy labels in training data and enhance the performance of DNNs in medical image analysis. Specifically, addressing the issue of noisy labels remains an ongoing challenge for the medical imaging analysis community. Dgani *et*

al. [21] adopted a noisy channel in neural networks, which models the stochastic relation between the correct label and the observed noisy label. Xue *et al*. [142] proposed an online uncertainty sample mining strategy to suppress the noisy samples. However, these methods do not distinguish mislabeled samples from hard samples. Making this distinction could greatly improve histopathology images analysis tasks with noisy labels.

In this work, we present a histopathology AL framework (PathAL) that is able to dynamically identify noisy labels and sample images that need to be annotated. Our goal is to provide a solution that is able to reduce annotations required from experts and to simultaneously handle noisy labels. For each iteration of PathAL, we first train the network using annotated images. We then force the network to modulate between overfitting and underfitting by adjusting the hyper-parameters. In this process, we monitor and rank the normalized average loss of every labeled sample and the normalized average predictive entropy of every unlabeled sample. We also measure the complexity of data points using their distribution density in the feature space and rank their complexity in an unsupervised manner. By doing so, the noisy labeled samples can be identified and discarded, while the hard and minority samples can be preserved. The unlabeled images that are most informative to the model are selected for annotations and added for training for next iteration. In addition, the typical unlabeled samples with the highest predictive confidence are added to the training pool with pseudo annotations generated by the model itself. This cost-effective sample selection strategy is able to improve the classification performance with far fewer manual annotations. Our proposed method is a tailor-made strategy for histopathology image analysis. The main contributions of this chapter include: 1) an AL framework (PathAL) that is able to dynamically identify important samples to annotate and to distinguish noisy from hard samples in the training set, 2) extensive experiments that demonstrate model improvement with less annotation effort and noisy samples; and 3) a reader study performed by a domain expert to validate our algorithm.

6.1.2 Related Work

6.1.2.1 Active Learning

A typical AL framework consists of a method to evaluate the *informativeness* of each unannotated data point x_u given $f'(x|L')$, where f' is a model trained on a labeled dataset L'. In literature, methods to evaluate *informativeness* can be generally classified into two types: 1) calculate the uncertainty, and 2) calculate the representativeness. In uncertainty-related methods, it is assumed that the more uncertain a prediction, the more information we can gain by including the ground truth for that sample in the training set. Wen *et al*. [137] proposed an AL method that uses uncertainty sampling to support quality control of nucleus segmentation in pathology images. Gal *et al*. [28] introduced Bayesian CNNs to measure the uncertainty of predictions. They demonstrated their approach for skin cancer diagnosis to show significant performance improvements over uniform sampling using the Bayesian Active Learning by Disagreement (BALD) method for sample selection [48], which sought to maximize the mutual information between predictions and model posterior. Konyushkova *et al*. [66] proposed to exploit geometric smoothness priors in the image space to aid the segmentation process in AL. They demonstrated state-of-the-art performance on mitochondria segmentation from electron microscopy (EM) images and on an magnetic resonance imaging (MRI) tumor segmentation task for both binary and multi-class segmentation. Another area of work focuses on the measure of representativeness in addition to uncertainty measures. This research uses the idea that methods only concerned with uncertainty have the potential to focus only on small regions of the distribution, and that training on samples from the same area of the distribution will introduce redundancy to the selection strategy or may skew the model towards a particular area of the distribution. Therefore, the selection method should also cover a large range of the data distribution in order to increase sample representativeness. Yang *et al*. [146] presented Suggestive Annotation, a deep AL framework for medical image segmentation, which uses an alternative formulation of uncertainty sampling combined with a form of representativeness density weighting. They demonstrated state-of-the-art performance using 50% of the available data on the MICCAI gland segmentation

challenge and a lymph node segmentation task. Smailagic *et al*. [122] proposed MedAL, an AL framework for medical image segmentation. They proposed a sampling method that combines uncertainty and distance between feature descriptors to extract the most informative samples from an unlabeled dataset. Ozdemir *et al*. [102] proposed a Borda-count based combination of an uncertainty and representativeness measure to select the next batch of samples. They introduced new representativeness measures such as "Content Distance," defined as the mean squared error between layer activation responses of a pre-trained classification network. Sourati *et al*. [124] proposed a method for ensuring diversity among queried samples by calculating the Fisher Information. They demonstrated the performance of their approach improved after labelling a small percentage of voxels, outperformed random sampling, and achieved higher accuracy than entropy based querying.

Our proposed PahtAL model combines both uncertainty and representativeness measures in the data selection algorithm. Unlike the methods discussed above, our AL framework also involves a *complementary sampling* strategy, in which the framework selects from an unlabeled dataset with: 1) a set of most uncertain samples to be annotated by an oracle, and 2) a set of highly certain samples that are "pseudo-labeled" by the framework. A similar idea has been proposed by [134] in natural images, but it has never been used in histopathology images. Furthermore, PathAL also considers the noisy label issue, which can deteriorate the performance of the AL framework in histopathology analysis. To the best of our knowledge, joint modeling of uncertainty and representation has not been explored in the previous literature in histopathology image analysis.

6.1.2.2 Noisy Label Detection

Addressing noisy labels in machine learning is an ongoing challenge. Several attempts have been made in natural image tasks. In general, there are two types of solutions to deal with noisy labels in a training dataset: 1) train models to detect the noisy labels and then clean or remove them to reduce their impact in the model training; and 2) directly train a noise-robust model with noisy labels. In line with the first approach, Koh and Liang [64] proposed an

influence function to measure samples that were "harmful" to model training. Lee *et al*. [78] proposed CleanNet, which was a joint neural embedding network. This approach summarized the knowledge of label noise from a fraction of manually verified classes. Transfer learning was then conducted to transfer the knowledge to other classes to handle label noise. Han *et al*. [43] proposed co-teaching, in which two deep networks were trained simultaneously. Each network selected which samples the other network used for training. Each of the networks taught the other to identify noisy labels. In [42], Guo *et al*. proposed CurriculumNet, in which training data were divided into several subsets by ranking their distribution density as a measure of complexity. The subsets were formed as a curriculum to teach the model to understand label noise gradually. A similar idea was proposed in [59]. In this work, a MentorNet was trained to identify potential noisy labels. The network then provided a data-driven curriculum for StudentNet, which was trained on the less noisy data samples. Huang *et al*. [50] proposed O2U-Net to make the network transition from overfitting to underfitting (O2U) automatically. By monitoring the training loss variation, they could detect and remove noisy labels from the original dataset. On the other hand, several other approaches that directly train a noise-robust model with noisy labels have been proposed. Goldberger and Ben-Reuven [35] proposed to model label noise by adding softmax layers to estimate the transition between correct labels and noisy labels. Xiao *et al*. [141] proposed a probabilistic model to describe the relations among images, true labels, noisy labels, and noise types. The probabilistic model required a small set of verified labels without noise. Reed and Lee [108] proposed the notion *consistent* to model noisy labels. Sample reconstruction errors were applied as the consistency objective to estimate the noise distribution. There are a few studies that have addressed issues of noisy labels in medical imaging. Dgani *et al*. [21] used a noise adaptation layer similar to [35] on a mammography classification task and outperformed standard training methods. Xue *et al*. [142] proposed an online uncertainty sample mining method (OUSM) to detect the noisy labels and iteratively re-weight sample losses.

To the best of our knowledge, we are the first to incorporate a noisy sample detector in

an AL framework. In our proposed PathAL, we adopt O2U-Net as a noisy label detector. It enhances our AL framework for the following reasons: 1) O2U-Net is a noise-cleansing method, so AL can be conducted after noisy label detection and removal to reduce the need for human annotations and improve the generalization capacity of the model; 2) other noise-cleansing methods require either particular assumptions on noise distribution estimation or extra specifically designed loss functions or networks (e.g. Co-teaching and MentorNet), while O2U-Net only requires adjusting the hyper-parameters of deep networks; and 3) by leveraging curriculum learning, in which images are divided into several subsets by ranking their distribution density in deep feature space, we can distinguish between the noisy labeled samples and the hard samples, which is a challenging task in histopathology image analysis.

6.1.3 Contributions

Our proposed method is a tailor-made strategy for histopathology image analysis. The main contributions of this chapter include: (1) an active learning framework (PathAL) that is able to dynamically identify important samples to annotate, distinguish noisy and hard samples in the training sets, is proposed; (2) extensive experiments are done to show promising results on enhancing the model performance with much less annotation efforts and noisy samples.

6.1.4 Organization

The rest of this chapter is organized as follows: we first discuss our proposed method in Section 6.2. The datasets used in our experiments and experimental results are shown in Section 6.3. Finally, conclusions are drawn in Section 6.4.

6.2 Methods

In this section, we first formally define our problem and the notations we use in this study. We then introduce curriculum sample classification and noisy sample detection methods, two key components of our proposed PathAL model. Finally we describe our proposed PathAL

method in detail.

6.2.1 Problem Definition

In traditional AL, we assume there is a large pool of unlabeled data U available and an oracle to help with labeling for every unlabeled data point x_u to add to labeled set L. We consider the whole training set to be $T = L \cup U = L_1 \cup U_1 = \cdots = L_k \cup U_k$, where L_i, U_i represents the labeled and unlabeled sets in ith iteration. AL starts from a small labeled set L_1 and tries to find the most informative samples $x^*_{1,j} \in U_1$. All the informative samples selected by the algorithm form a set I_1 and will be annotated by domain experts and added to the labeled set for model training in the next iteration. Thus, we have $L_2 = L_1 + I_1$ and in general $L_{i+1} = L_i + I_i$.

In contrast to the traditional AL model, our proposed PathAL considers three groups of samples in the data pool: 1) annotated samples that are in the training dataset that have a high probability of incorrect label assignment (noisy samples), denoted as the noisy set N_i; 2) unlabeled samples that are most informative to the current model (informative samples), denoted as the informative set I_i; and 3) unlabeled samples for which the current model is confident in its predictions (confident samples), denoted as the confident set C_i. PathAL will discard the noisy samples, require experts to annotate the informative samples and add them to the training pool, and add confident samples to the data pool with their own, model-assigned annotations, simultaneously. Thus we have $L_{i+1} = L_i - N_i + I_i + C_i$, where $N_i \subseteq L_i$, $I_i \subseteq U_i$, and $C_i \subseteq U_i$, and the training set $T = L_i + U_i$. The general process of PathAL is illustrated in Figure 5.1(a). The core goals for PathAL are: 1) detect the noisy samples and distinguish them from hard samples, and 2) detect the informative samples to be annotated and add confident samples automatically. To meet these goals, we first briefly discuss our curriculum sample classification method, inspired by CurriculumNet [42] and O2U-Net [50] noisy sample detection, upon which these two questions are answered in PathAL.

6.2.2 Curriculum Sample Classification

A key component of our PathAL framework is to leverage curriculum learning to classify each example in a training set to be easy, medium and hard based on its complexity. We extend CurriculumNet [42] for AL scenarios and use it in a fully unsupervised fashion. In each iteration, we use a trained model to compute a deep representation for each image in the training set T. This step aims to roughly map all training images into a feature space where the underlying structure and the complexity of the images can be discovered. We then classify each sample into different complexity levels, ranging from easy samples with high-signal labels to difficult samples whose labels may contain noise. To do so, we first reduce the dimension of the deep features using t-distributed Stochastic Neighbor Embedding (t-SNE) [91]. With this set of reduced features, we use the K-means algorithm to cluster the images into different groups. Each group will ideally contain images with similar diagnoses. This step aims to help the following process select representative samples covering the whole training sample space. Next, we calculate a Euclidean distance matrix $D \subseteq \mathbb{R}^{n \times n}$ as,

$$D_{i,j} = \|f(I_i) - f(I_j)\|^2 \tag{6.1}$$

where n is the number of images in the same group, I_i, I_j are two images in this group, $f(I_i), f(I_j)$ are the feature vectors of the two images in deep feature space. $D_{i,j}$ indicates a similarity value between I_i and I_j. Then we calculate a local density (ρ_i) for each image,

$$\rho_i = \sum_j X(D_{i,j} - d_c) \tag{6.2}$$

where

$$X(d) = \begin{cases} 1 \ d < 0 \\ 0 \ \text{other} \end{cases} \tag{6.3}$$

d_c in the above equation is a distance threshold we select for determining the local density. It is selected by first sorting n^2 distances from small to large values, and choosing the top $k\%$. Following the practice in [42], we set $k = 60$ in all our experiments. The local density ρ_i counts how many samples are closer to image I_i in the deep feature space than the threshold d_c. Finally, we use a K-means clustering method to classify each sample as easy, medium,

or hard based on their local density for each group. To this end, we assume that a group of easy images with correct labels will often have similar visual characteristics, project closely to each other in the feature space, and therefore have a high ρ_i. By contrast, hard images often have more visual diversity, resulting in a sparse distribution with a smaller ρ_i. Figure 5.1(b) illustrates the workflow of our curriculum classification (CC) algorithm. We also summarize the CC in Algorithm 4.

Algorithm 4 Curriculum Classification

Require: trained DNN (f), images (x_i) in training set T;

(1) Generate deep image features $f(x_i)$ for each image x_i;

(2) Reduce dimensionality using t-SNE, then use K-means cluster algorithm to cluster these features into k different groups $g_1, g_2, ...g_k$;

(3) Calculate a Euclidean distance matrix $D \subseteq R^{n \times n}$ as $D_{i,j} = \|f(x_i) - f(x_j)\|^2$;

(4) Calculate a local density function for each image $\rho_i = \sum_j X(D_{i,j} - d_c)$;

(5) Use K-means cluster algorithm to classify each image x_i to easy, medium, and hard based on their local density ρ_i in each group g_i.

Note that although our curriculum classification algorithm is inspired by CurriculumNet [42], it is substantially different from it in the following aspects. CurriculumNet is performed in weakly supervised learning settings, where the authors have access to all the labels of the samples and are able to use the subgroup with same label for curriculum classification. In this study, our curriculum classification is performed in each iteration of AL, where we do not have full access to the annotations of training samples. Therefore, we have to use unsupervised K-means clustering to first group the training images, and then classify the samples in each group based on the local density. In this way, we are able to sample the unlabeled images evenly in the deep feature space.

As our model evolves during each AL iteration, it is hard to distinguish whether a sample is a noisy sample that has a wrong label or is a complex sample that the model has not learned yet. Accordingly, we introduce another key component of PathAL, with which we are able to distinguish noisy samples from hard ones, and discover the most informative samples to be annotated.

6.2.3 Noisy Sample Detection

It is challenging to determine whether an incorrectly classified sample is a noisy one with the wrong label or a complex one that is inherently hard to learn for deep learning models. The CC algorithm in Section 6.2.2 only considers the visual complexity of the training samples, but does not provide much information about how well the current AL model learns these samples. To help with this, we introduce a noisy sample detector by using O2U-Net [50].

The key observation from O2U-Net is that noisy-labeled samples are usually memorized at the late stages of training, as is the case with hard samples. At the beginning of training, when the network is still underfitting, the losses of noisy and hard samples are larger than those of easy samples because the model quickly fits to easy samples. Conversely, during the late stages of training, the network usually overfits to the training set. It memorizes both the noisy/hard samples and easy samples, so that the losses generated from them are indistinguishable. Therefore, by tracking the variation of loss for every sample at different stages of training, it is possible to detect noisy and hard samples. Based on this idea, the O2U-Net attempts to cycle training between underfitting and overfitting by tuning the learning rate, while observing the variation of loss for every sample in L_i. Specifically, at the beginning of training, a large learning rate is set. The learning rate gradually decreases to some extent during training and is then reset to the original learning rate. This process repeats for multiple rounds until enough loss statistics are gathered. When the network almost converges to some minimum (nearly overfitting), a large learning rate can make the network jump out of the minimum. As a result, the network will quickly start underfitting the data. By monitoring the training loss for each sample, we can expect the larger the average loss of a sample after the cyclical training, the higher probability of being a mislabeled or a hard sample. We apply the same network to detect noisy labels and to train the final classifier using EfficentNet-B0 [131] (see Section 6.3.3 for more training details). For a more detailed description of O2U-Net, please refer to [50].

The original O2U-Net only monitors the training loss for each sample in L_i. We extend it to monitor the predictive entropy for every sample in the unlabeled dataset U_i. Specifically,

we do inference after each epoch in the O2U training cycles. We record the predictive entropy for each sample in U_i and find the samples with highest average predictive entropy. These samples are the most "informative" samples to the current model because they cannot be predicted confidently and may not be represented in the feature space of the current labeled set L_i. We summarize the O2U training workflow in Algorithm 5. We point out that the O2U-Net alone cannot distinguish between noisy samples and hard samples. With the help of curriculum classification, however, we are able to heuristically separate these two types of samples, which we will discuss in the next section.

Algorithm 5 Training O2U

Require: trained DNN (f), labeled image x_{l_i}, unlabeled image x_{u_i};

for Each epoch **do**

Adjust learning rate via Equation (6.8).

for Each labeled image x_{l_i} **do**

(1) Compute and record training loss lss_i;

(2) Update the network f;

end for

for Each unlabeled image x_{u_i} **do**

(1) Compute and record predictive entropy ent_i;

end for

end for

(1) Compute the normalized average loss $\overline{lss_i}$ of every labeled sample among all the epochs;

(2) Compute the normalized average predictive entropy $\overline{ent_i}$ of every unlabeled sample among all the epochs;

(3) Obtain the order by ranking all the labeled samples by $\overline{lss_i}$ and all the unlabeled samples by $\overline{ent_i}$.

6.2.4 PathAL

After introducing the CC algorithm and the O2U process, we now specify the core goals of PathAL: 1) detect noisy samples and distinguish them from hard samples, and 2) detect informative samples to be annotated and add confident samples using "pseudo-labeles" assigned by the model itself.

At the ith iteration of PathAL, we first train a network using the current labeled dataset L_i until it converges. Then we apply the O2U process to continue the training. We monitor the loss variation of each labeled sample and predictive entropy of each unlabeled sample. Simultaneously, we apply the CC algorithm on the samples in training set T and classify them into easy, medium, and hard samples based on their local density in the feature space. To detect the noisy samples, we find those that have large loss variations in L_i and are also classified as easy by the the CC algorithm. On one hand, these samples have large loss variations, which means they are hard to learn by the current network. On the other hand, the samples must have a high local density in the deep feature space in order to be labeled as "easy", *i.e.* they are typical samples that are visually similar to other samples in T. Thus, there is a higher probability that the pathologist annotations for these samples contain noise. To prevent them from impacting the model's training and performance, we discard these samples in the next training iteration. To detect the informative samples that require additional expert annotations, we select the samples with the highest average predictive entropy during O2U training. As discussed in Section 6.2.3, these samples are most informative because they cannot be predicted confidently by the current model. In addition, we add unlabeled samples that have the lowest predictive entropy and are classified as "easy" or "medium" by the CC algorithm. Our model is confident in these predictions, and they are "typical" samples in the deep feature space, so there is a high probability that the model predictions are correct. Therefore, it is cost-effective to add them automatically into L_{i+1} with self-assigned "pseudo-labels." Algorithm 6 illustrates the workflow of our PathAL algorithm.

Algorithm 6 PathAL

Require: a DNN (f), training set T, initial selected labeled image set L_1 and the rest unlabeled image set U_1;

for Each iteration of PathAL **do**

(1) Train DNN f based on the current labeled image set L_i until it is converged;

(2) Perform CC and O2U training in training set T;

(3) Select noisy samples N_i, most informative samples I_i, and confident predicted samples C_i based on CC and O2U results;

(4) Update the training set for next iteration as $L_{i+1} = L_i - N_i + I_i + C_i$;

end for

As the model evolves during the training process, both the CC and O2U results change. Therefore, we do not discard the noisy samples completely. Instead, we keep them in a pool and examine if they need to be added back in throughout the training process. In doing so, we build a mechanism for the model to correct errors made at the beginning of the AL process. We summarize how to combine the CC and O2U results based on their relationship in Figure 5.1(c).

6.3 Datasets, Experiments and Results

In this section, we first introduce the dataset and evaluation metrics we used in our experiments. We then discuss the implementation details of our model, followed by the description of several baseline models. Finally, we demonstrate and discuss the experimental results.

6.3.1 Dataset and Pre-processing

To demonstrate the effectiveness of our PathAL technique, we use the Kaggle dataset from the "Prostate cANcer graDe Assessment using the Gleason grading system" (PANDA) challenge to simulate the AL scenario. The dataset consists of over 11,000 whole-slide images of digitized H&E-stained biopsies originating from two centers (Karolinska Institute and Radboud University Medical Center). Different slide scanners with slightly different maximum microscope resolutions were used for digitization and labels were generated from different

pathologists. The Karolinska dataset was labeled by a single experienced pathologist. Label noise may exist in this dataset due to the lack of label validation by another pathologist. The Radboud dataset was read by trained students. For this dataset, some minor label noise may also exist in the training set due to mistakes in the annotation process or inconclusive results. Though the label noise presents a modeling challenge, it resembles many real-world scenarios. As mentioned above, even experts in the field with years of experience do not always agree on how to interpret prostate histology.

Each sample image in the PANDA dataset is a large, which requires an efficient algorithm to locate areas of concern on which to focus. We used our previously developed tiling algorithm with a blue-ratio selection criteria to identify the most informative tissue areas [126]. Specifically, the algorithm consists of four steps. First, a binary mask of the tissue on the slide is created by setting a threshold for the average intensity. This threshold is set empirically to 90% of the maximum image intensity value. Second, the mask is smoothed using morphological closing and the skeleton of the smoothed mask is then found and branches are removed by finding the endpoints with the maximum geodesic distance. Third, the mid-line is partitioned based on the patch size and overlap, tangent lines are found at each of these locations by looking at the neighborhood of nine pixels along the mid-line and the perpendicular line is drawn until intersection with the mask boundary. Finally, a set of patches that intersect with more than 60% with the mask are chosen to calculate their blue ratio, and the top k blue-ratio patches are selected. In this work, a patch size of 256×256 pixels was used, and 36 patches were selected for each slide. Figure 5.4 illustrates the pre-processing steps of the PANDA dataset.

6.3.2 Evaluation Metrics

The task of the PANDA challenge is to predict the ISUP grade on a 0-5 scale for each biopsy image based on Gleason grading system. Gleason grading is a subjective task, with high inter- and intra-observer variability. Agreement between pathologists is often measured using Cohen's kappa. Therefore we used the quadratic weighted kappa (QWK) to evaluate

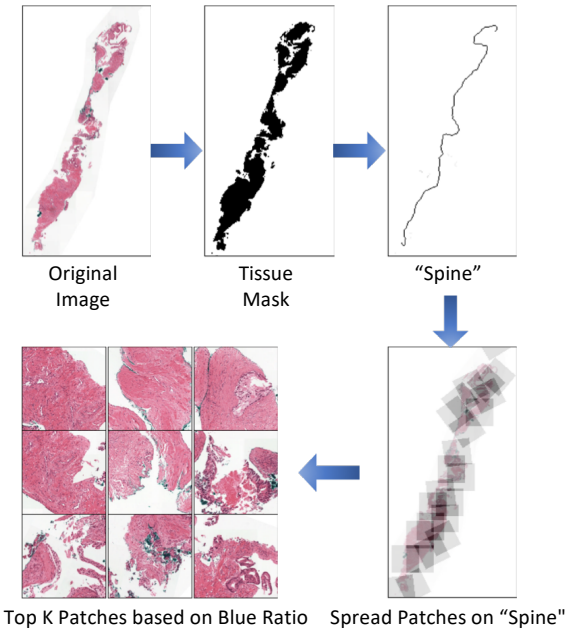

Figure 6.2: Illustration of data pre-processing steps. A binary mask of tissue is first extracted; then the mid-line is found using morphological closing; after that, the mid-line is partitioned to form patches based on the batch size and overlap; finally, the blue ratios of patches are calculated and the top k patches are selected.

our model's performance. QWK measures the agreement between two outcomes. It typically varies from 0 (random agreement) to 1 (complete agreement), though it may be negative if there is less agreement than expected by chance.

The QWK is calculated as follows. First, an $N \times N$ histogram matrix O is constructed, such that $O_{i,j}$ corresponds to the number of ISUP grade i (actual) that received a predicted value j. An $N \times N$ matrix of weights, w, is calculated based on the difference between actual and predicted values as,

$$w_{i,j} = \frac{(i-j)^2}{(N-1)^2} \tag{6.4}$$

After that, an $N \times N$ histogram matrix of expected outcomes, E, is calculated assuming that there is no correlation between values. This is calculated as the outer product between the actual histogram vector of outcomes and the predicted histogram vector, normalized such

that E and O have the same sum. From these three matrices, the QWK is calculated as,

$$\kappa = 1 - \frac{\sum_{i,j} w_{i,j} O_{i,j}}{\sum_{i,j} w_{i,j} E_{i,j}} \tag{6.5}$$

6.3.3 Network Backbone, Loss Function, and Other Training Details

In this study, we used EfficientNet-B0 [131] for all of our experiments. We used EfficientNet-B0 because it achieved competitive results in the challenge without high computational cost. Note that PathAL does not require a specific network backbone and can be easily adapted to use other networks in various scenarios. We used normal Adam optimization in all the experiments. The model is trained on one single Tesla V100S GPU in PyTorch.

To predict an ISUP grade on a 0-5 scale, we used the ordinal regression loss function at the final layer of our network. This can better capture the ordinal relationship between grade and severity in the training set compared to multi-class classification or the mean square error loss function. Specifically, we used binary cross entropy loss with binning labels. For example, label = $[0, 0, 0, 0, 0]$ means ISUP grade 0, label = $[1, 0, 0, 0, 0]$ means ISUP grade 1, and label = $[1, 1, 1, 1, 1]$ means ISUP grade 5.

We performed four-fold cross-validation to show the effectiveness and robustness of PathAL. In each fold, we used a hold-out set as the testing set, and the rest as the training set. For comparison, we asked the expert pathologist to annotate 10% of the training samples each time. We compared PathAL performance with other baseline models alongside the pathologist annotations. In each iteration of PathAL, we excluded 1% of the whole training set $|T|$ as noisy samples, added the other $10\% * |T|$ annotated data and $5\% * |T|$ confidently predicted samples with their "pseudo-labels."

6.3.4 Baselines

We compared our model with three baseline models. In this section, we describe the acquisition functions used by these baseline models.

- Choose pool points that maximize the predictive entropy (Max Entropy). As the

ordinal regression used in our experiments can be viewed as a binary classification in each position of the output, we can form the entropy calculation as,

$$\sum_{i=1}^{5}[-p_i \log(p_i) - (1-p_i)\log(1-p_i)] \tag{6.6}$$

- Choose pool points that are predicted with low confidence (also known as variation ratios). In our ordinal regression formation, a simple way to represent the prediction confidence can be calculated as,

$$\sum_{i=1}^{5}|p_i - 0.5| \tag{6.7}$$

- Choose pool points randomly: $a(x) = unif()$ with $unif()$ as a function returning a draw from a uniform distribution over the interval [0, 1]. Using this acquisition function is equivalent to choosing points uniformly at random from the pool.

6.3.5 Experimental Results

We conducted experiments in various settings and compared PathAL with other AL baselines. Note that the ISUP grade for samples in U_i is not available in the real AL scenario. However, we used the label in the dataset as ground truth to provide a quick sanity check and demonstrate that PathAL worked as expected.

6.3.5.1 Illustration of Curriculum Classification

We first illustrate the process of the CC algorithm to help explain its effectiveness. As training proceeds, the deep image features should be more separable according to their ISUP grades in feature space. In other words, if we use k-means to group them in a fully unsupervised fashion, the label diversity within one group should decrease. We defined a metric called "grade concentration" to measure the ISUP diversity for each cluster group at each iteration in PathAL. The "grade concentration" was calculated as an average negative entropy of ISUP grade distribution of each group. Figure 6.3(a) demonstrates the t-SNE plot

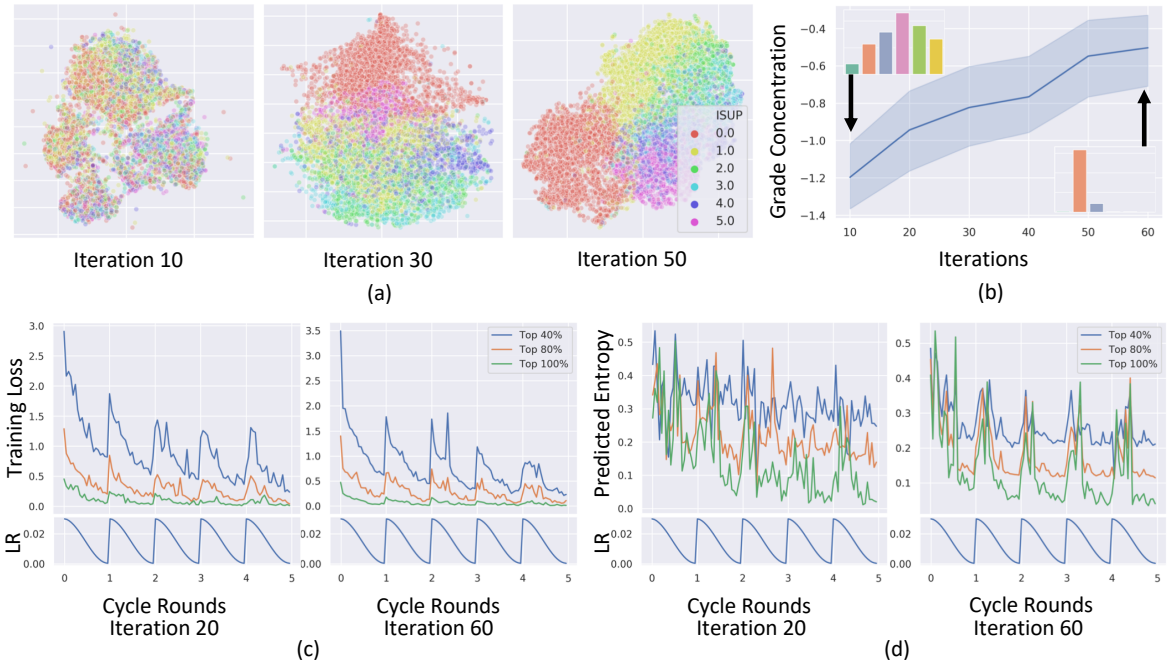

Figure 6.3: (a) t-SNE plot in deep feature space. Each point in the figure represents a slide whose color indicates its ISUP grade. As training went on, different ISUP grades became more separable in the deep feature space, indicating the model captured more essential information to make the correct predictions. (b) The trend of "grade concentraion" that measured the ISUP grade distribution within subsets clustered by k-means. The insets of the figure demonstrates a typical ISUP distribution for the subsets. At the beginning of training, the ISUP grades were more diffuse, while at the end of the training, each cluster concentrated on fewer grades. (c)(d) The training loss for every sample in L_i, and predictive entropy for every sample in U_i during the O2U process.

in the deep feature space. Each point in the figure represents a slide whose color indicates its ISUP grade. As expected, in the early iterations of PathAL, different ISUP gardes were not well separated in the feature space, indicating the model was not able to achieve high accuracy in its predictions. As training went on, ISUP images became more separable according to their grades. As a result, the subsets clustered by k-means methods would have higher "grade concentration". Figure 6.3(b) depicts the trend of "grade concentraion." The insets of the figure demonstrate a typical ISUP distribution for the clusters. At the beginning of the training iterations, the ISUP grades were more spread, while at the end of the training, each cluster had lower label diversity.

6.3.5.2 Illustration of O2U Cyclic Training

In this illustration, we demonstrate the cyclic training in O2U process. After the model converged in each iteration, we adjusted the learning rate periodically so that the network could transition from overfitting to underfitting cyclically. The learning rate was adjusted based on a cosine annealing function in each cyclic round as,

$$lr = lr_{min} + \frac{1}{2}(lr_{max} - lr_{min})(1 + \cos(\frac{T_{cur}}{T_{max}}\pi)) \tag{6.8}$$

where T_{max} was the epoch for one cycle, lr_{min} and lr_{max} were the minimum and maximum learning rates in one cycle.

We monitored the training loss for every sample in L_i, and the predictive entropy for every sample in U_i. This process is illustrated in Figure 6.3(c)-(d). After the cyclic training, the samples were ranked according to their losses and predictive entropy variation. The samples were plotted in terms of three groups: top $0\% - 40\%$ ranked samples, top $40\% - 80\%$ ranked samples, and the rest of the samples. It was observed that the training losses and predictive entropy fluctuated with the cyclical adjustment of the learning rate. The training losses of the top 40% of samples fluctuated drastically during the cyclical training when compared to the rest of the samples, which may indicate they were noisy or hard samples (see Figure 6.3(c)). It is also observed in Figure 6.3(d) that the top 40% samples ranked for predictive

entropy did not change during early iterations, which implies that the samples in L_i contained limited information for the model to classify these samples. As the training proceeded, we observed that the predictive entropy of the top 40% started to fluctuate, as the samples in L_i contained more information about the samples in U_i, even for the most uncertain group.

6.3.5.3 PathAL Performance

We compared our proposed PathAL with the three common AL baselines mentioned in Section 6.3.4. Figure 6.4(a) demonstrates the comparison results of QWK on ISUP grade prediction between PathAL and the baselines. The QWK was calculated as the average performance for the four folds and the standard deviation was plotted as the error bar. As shown in the figure, PathAL significantly improved the QWK with less required annotations. It achieved a higher QWK compared with the full training set supervision baseline with only 60% annotations required for the expert. The lowest confidence and predictive entropy methods performed better at the early stage of AL (when the annotated samples were limited). However, their effectiveness gradually decreased and converged to the same level of the fully supervised performance when the model had access to the full annotations. Though PathAl did not outperform the fully-supervised baseline by a large margin, we argue that it achieved slightly better performance because it discarded the noisy labels. We also found that the predictive entropy model achieved its highest QWK performance when using only 90% annotations, indicating that excluding "noisy" samples can improve the prediction accuracy. To illustrate the effectiveness of each component in PathAL, we performed an ablation study in Section 6.3.5.4.

To determine whether: 1) the samples we discarded in each iteration were noisy with low QWK, 2) the samples we asked the expert to annotate were "informative" with low QWK, and 3) the samples we added with their "pseudo-label" were correct with high QWK, we plotted the QWK for each group during the training process in Figure 6.4(b). It is observed that the noisy sample group N_i had a relatively low QWK even though their labels were used for training, while the confident predicted samples in U_i have a much higher QWK. Samples

in I_i that were re-annotated by the expert had low QWK, indicating that those samples were most "informative" to the current models, and would improve the model's performance by a large margin if added to L_i with annotations.

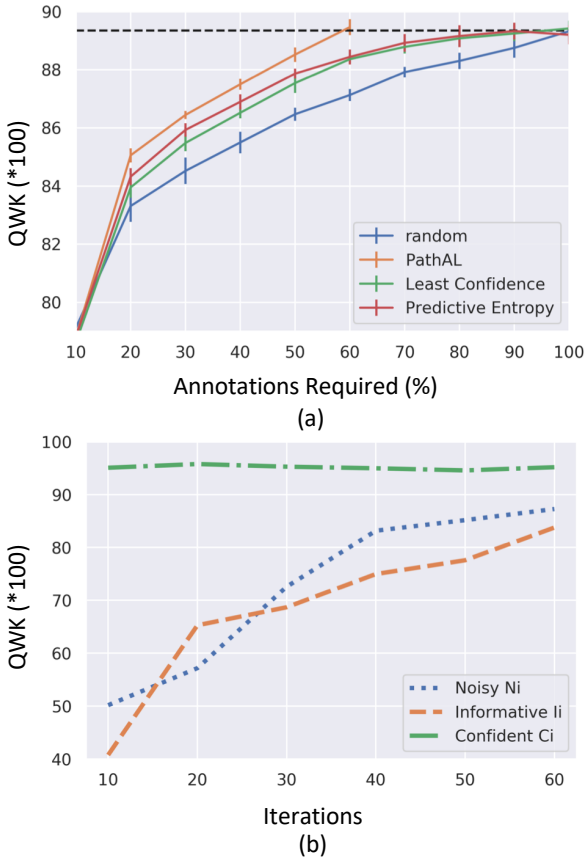

Figure 6.4: (a) Performance comparison between PathAL and other AL baselines. (b) QWK for each group (N_i, C_i, I_i) during the training process.

6.3.5.4 Ablation Study

To illustrate the effectiveness of each component in PathAL, we performed an ablation study with varying amounts of labeled data until 60% of expert annotations were added to L_i, when PathAL would have access to all sample labels either through expert's annotations or pseudo-assigned labels. Section 6.3.5.4 demonstrates the results when only parts of PathAL

components were used. The QWK is shown as an average for four different folds with a calculated standard deviation.

From the table, we observed that using a simple predictive entropy measure by O2U in row 2 (Entropy (O2U)) to select the most "informative" samples improved the QWK by 1.3% compared with the random selection baseline, while PathAL improved the random baseline by 2.4%. When we excluded the noisy samples in row 3 (Entropy + Noisy (O2U)), we saw QWK improve by 0.3%, while adding the high-confidence predictive samples by self-assigned pseudo-labels (row four Entropy + Conf Preds) improved the QWK by 0.4%. To illustrate whether the O2U component helped with the selection of N_i, I_i, C_i, we implemented PathAL with selection based on the predictive entropy (row 5 PathAL (w/o O2U & CC)), *i.e.* we selected the top ranked samples in L_i based on the model's predictive entropy as N_i, the top ranked samples in U_i as I_i, and the bottom ranked samples as C_i. We showed that using O2U and CC in combination as a sample selective strategy improved the QWK by 1.3%, indicating the effectiveness of their roles in PathAL.

Table 6.1: Ablation Study of PathAL.

	QWK
Random	87.1 ± 0.6
Entropy (O2U)	88.4 ± 0.5
Entropy + Noisy (O2U & CC)	88.6 ± 0.3
Entropy + Conf Preds (O2U & CC)	88.8 ± 0.5
PathAL (w/o O2U & CC)	88.6 ± 0.4
PathAL	**89.5 ± 0.5**

6.3.5.5 Pathologist Validation

To validate our algorithm for detecting easy, noisy, and hard samples, a pathologist at our institution with expertise in Gleason grading (AS) performed an independent reader study. Specifically, we provided three groups of slides that were labeled as easy, noisy, and hard by

our algorithm. Each group consisted of 100 slides. The pathologist was asked to give final ISUP grades without knowing the ground truth labels provided by the dataset. As shown in Section 6.3.5.5, we measured the QWK of each group. Surprisingly, the QWK of the "easy" group was 1, indicating 100% agreement between the pathologist and the ground-truth label for the 100 samples. Conversely, the QWK of the "noisy" group was -0.14. The high variance between the dataset labels and pathologist readings may be explained by noise in the dataset labels. The QWK of the "hard" samples was 0.10, which was slightly higher than that of the "noisy group". By visually inspecting the discordant slides in the "hard" group, we found that some of the slides were likely to have incorrect labels. In other slides, the cancerous regions were either small or ambiguous, so it was difficult for the pathologist to spot the cancerous areas or reach a consensus. In general, we found that our algorithm robustly distinguished between "easy" and "hard & noisy" groups. However, there is still room for improvement in distinguishing between "hard" and "noisy" samples. For more details, please refer to the Appendix.

Table 6.2: Pathologist Reader Study.

Group	Easy	Hard	Noisy
QWK	1.00	0.10	-0.14

6.4 Conclusion

In this chapter, we have proposed PathAL, a novel AL framework for histopathology image analysis. Unlike prior studies in medical image AL, which only consider the most "informative" samples to be added in each iteration, PathAL also heuristically excludes noisy samples and adds confident predictive samples with self-assigned pseudo-labels. Specifically, the combination of a curriculum classification (CC) algorithm and an overfitting to underfitting (O2U) process was used to detect noisy, confident and informative samples. Our proposed method achieved competitive performance while requiring only 60% of samples to

be annotated, compared to the fully supervised learning baseline on the PANDA challenge. Extensive experiments conclude the effectiveness of each component in PathAL. We expect to apply PathAL to other histopathological image analysis scenarios in the future.

CHAPTER 7

Conclusion

7.1 Summary of Contributions

In this dissertation, we focus on one of the biggest challenges in histopathology image analysis using deep learning, namely the insufficient number of labeled images for training. We have designed and analyzed novel deep learning models to enable cost-effective, scalable image processing and diagnosis on histopathology in supervised and semi-supervised settings. Furthermore, we have studied an active learning framework, which is also known as "human in the loop" approach, to further reduce the experts' annotation effort. Our proposed active learning framework is tailored to the specific characteristics in histopathology image analysis.

In Chapter 2, we start with a fully supervised segmentation method, which is named as Path R-CNN, for Gleason grading of prostate cancer. We formally define our problem in the context of the image instance segmentation problem. We assign the stromal components of the input images as the background class. Other epithelial cells in the input image that have been annotated by the pathologists as benign, low-grade or high-grade are assigned as instance objects, *i.e.* the RoIs we want our network to find. Under these assignments, we take advantage of R-CNN model and modify it to be more suitable for the Gleason grading task. Specifically, we use ResNet as the backbone for our image parser. First, the image parser generates feature maps. These feature maps are then fed into two branches. In one branch, we adopt the same two-stage procedure as in the Mask R-CNN. The feature maps are first used by a Region Proposal Network (RPN) that generates region proposals (RoIs). In the second stage, a Grading Network Head (GNH) is then used for predicting the class, box offset, and a binary mask for each RoI. To this we add another branch that outputs an

epithelial cell score that detects the presence of epithelial cells in the image. We refer to this part as the Epithelial Network Head (ENH). The final prediction of the network depends on the results of the ENH and GNH. Finally, a post-processing step based on a conditional random field is applied to the prediction.

The main contributions of our proposed Path R-CNN are twofold: first, by adding an Epithelial Network Head (EHN), we adapt the Mask R-CNN to be suitable for the histological image analysis for Gleason grading task with little additional computational overhead; second we develop a two-stage training strategy which enables our model to detect epithelial cells and predict Gleason grades simultaneously.

In Chapter 3, Chapter 4, and Chapter 5, we present a series of studies focusing on semi-supervised learning (SSL) using generative adversarial networks (GAN). We first focus our efforts on natural images. In Chapter 3, we systematically compared two GAN-based SSL methods, Good GAN and Bad GAN, by applying these two models with commonly-used benchmark datasets. We illustrate the distinct characteristics of the images they generated, as well as each model's sensitivity to varying the amount of labeled data used for training. In the case of low amounts of labeled data, model performance is contingent on the selection of labeled samples; that is, selecting non-representative samples results in generating incorrect image-label pairs and deteriorating classification performance. Furthermore, selecting the optimal batch size is crucial to achieve good results in both models. Notably, Good GAN and Bad GAN models can be used for complementary purposes; Good GAN generates good image-label pairs to train the classifier, while Bad GAN generates samples that force the decision boundary between data manifold of different classes. Inspired by this study, in a follow-up study we present in Chapter 4, we develop a unified-GAN (UGAN), a novel framework that enables a classifier to simultaneously learn from both good and bad samples through adversarial training. We perform extensive experiments on various datasets to show that UGAN: 1) achieves competitive performance among other GAN-based models, and 2) is robust to variations in the amount of labeled data used for training. Overall, our main contributions of this study are: 1) we propose a novel SSL framework, UGAN,

which simultaneously trains a good and a bad generator through adversarial training and takes advantage of both generated samples to boost SSL performance; 2) we analyze our proposed UGAN, theoretically prove its global optimum, and additionally put UGAN in the Expectation-Maximization (EM) framework and validate its non-increasing divergence property; and 3) we do extensive experiments to show that UGAN can improve upon Triple-GAN and Bad GAN classification results in SSL, and show the effectiveness of the model with different amounts of labeled data.

In Chapter 5, we switch our gear back on histopathology image analysis. Our goal is to synthesize a realistic histopathology image x based on an arbitrary semantic mask y, so that (x, y) can be used to compensate for the small data size when training a segmentation network. Image synthesis for data augmentation is not widely used in histopathology analysis because generating images with fine details is difficult. Synthesizing images for gland segmentation poses even more challenges, as the generated images have to preserve both global gland structures and finer nuclear details based on the masks. To overcome these problems, we design the generation and segmentation networks using pyramid structures. We show that the synthesized image-mask pairs can be used to boost the segmentation performance, especially in semi-supervised scenario. The main contributions of our model are twofold. First, by using a pyramid generation scheme, we are able to generate large-scale histopathological images up to 1024×1024 at high resolution (20x). Compared to the state-of-the-art pathology synthesis methods, which generate images up to 256×256 allowing for only limited context such as simple nuclei, our generation allows to incorporate richer context such as gland structures and nuclei details that are useful for precise diagnosis. Second, the generation is based upon a conditional method, which produces good image-mask pairs. These image-mask pairs can be used to compensate for the lack of data points in training segmentation models. We demonstrate the effectiveness of our method in segmentation tasks and analyze how it performs differently in supervised and semi-supervised settings.

In Chapter 6, we study an active learning framework that is tailored to histopathology image analysis, namely PathAL. PathAL is able to dynamically identify the noisy labels and

sample the images that need to be annotated. We provide a solution that is able to reduce the annotations required from the expert and handle noisy labels simultaneously. Specifically, for each iteration of PathAL, we first train the network using the annotated images. Then we make the network to transfer from overfitting to underfitting status cyclically by adjusting the hyper-parameters. In this process, we monitor and rank the normalized average loss of every labeled example and the normalized average prediction entropy of every unlabeled example. We also measure the complexity of data points using their distribution density in the feature space, and rank their complexity in an unsupervised manner. By doing so, the noisy labeled samples can be identified and discarded, while the hard and minority samples can be preserved; the unlabeled images that are most informative to the model as it trains are selected for annotations and add to the training for the next iteration. In addition, the typical unlabeled samples with highest predictive confidence are added to the training pool with pseudo annotations generated by the model itself. This cost-effective sample selection strategy is able to improve the classification performance with much less manual annotations. Our proposed method is a tailor-made strategy for histopathology image analysis. The main contributions of this work include: 1) an active learning framework (PathAL) that is able to dynamically identify important samples to annotate, distinguish noisy and hard samples in the training sets, is proposed; 2) extensive experiments are done to show promising results on enhancing the model performance with much less annotation efforts and noisy samples.

7.2 Future Works

There are still many open research problems in the topics that this dissertation doesn't cover.

7.2.1 Discovery of Novel Objects in Long-tail Distribution

In real diagnostic situations, unexpected objects could exist. For example, aberrant organizations and rare tumors can show up in inference time while they are not included in training data. Although this dissertation focuses on how we can use data effectively, we didn't put

much effort on dealing with the novel objects in the long-tail distribution. In the future, we can potentially solve this problem by adding an outlier detection algorithm to our proposed models. The outlier detection algorithm can raise a red flag whenever it sees a novel object, so that domain experts can take a closer look at it.

7.2.2 Correlate Deep Features with Clinical-Relevant Features

In Chapter 5, we provide image manipulation results by changing the gland labels in Figure 5.4(c). It suggests that deep learning models are able to learn useful representations that are predictive for the clinical-relevant measurements. Nevertheless, in this dissertation we don't provide any quantitative analysis on the learned representations. In the future, a potential direction is to make the latent representations more explainable and correlate them with clinical-relevant measurements through mutual information maximization. It is also helpful to seek assistance from pathologists to provide clinical-relevant measurements.

7.2.3 Interpretable Deep Learning Models

Deep learning is often criticized as a "black box" for its decision-making process, since it is not understandable to humans. In histopahtology image analysis, doctors and patients want to know the decision process for the diagnostic basis. In Chapter 6, we present a heuristic way to distinguish between noisy samples and hard samples, and provide interpretability to some extent. Nevertheless, the explanation is purely from the computational point of view and doesn't incorporate opinions from domain experts. Thus a crucial future direction is to increase the interpretability of our developed models.

APPENDIX A

Appendix for Chapter 2

A.1 Gleason Grading System for Prostate Cancer Diagnosis

The most common method for histological grading of prostate tissue is the Gleason grading system. Depicted in Figure A.1, this system classifies tissue into five grades, numbered 1-5. The grade increases with increasing malignancy and, therefore, cancer aggressiveness. Gleason grade characterizes tumor differentiation, *i.e.*, the degree of tumor resemblance to normal tissue. Grade 1 corresponds to well differentiated tissue, *i.e.*, tissue with the highest degree of resemblance to normal tissue, and indicates a high chance of patient survival. Grade 5 corresponds to poorly differentiated tissue and indicates a lower chance of survival.

In this chapter, we classify tissue into four categories based on the Gleason grading results. These are Stroma (ST), the fibromuscular tissue surrounding glands; Benign (BN), tissue featuring well-formed glands, which are rated as Gleason 2 or below; Low-grade (LG), tissue featuring recognizable glands with darker cells, which are rated as Gleason 3; and High-grade (HG), tissue featuring non-recognizable, poorly differentiated glands, which are rated as Gleason 4-5.

A.2 More Insights for ENH and Comparison with Multi-Scale U-Net

The starting point of our chapter is our previous Multi-scale U-Net work in [79]. We found the main drawback of the Multi-scale U-Net model is the softmax layer at the end. As illustrated in Figure A.2, the Multi-scale U-Net model performs much better than the U-Net; however

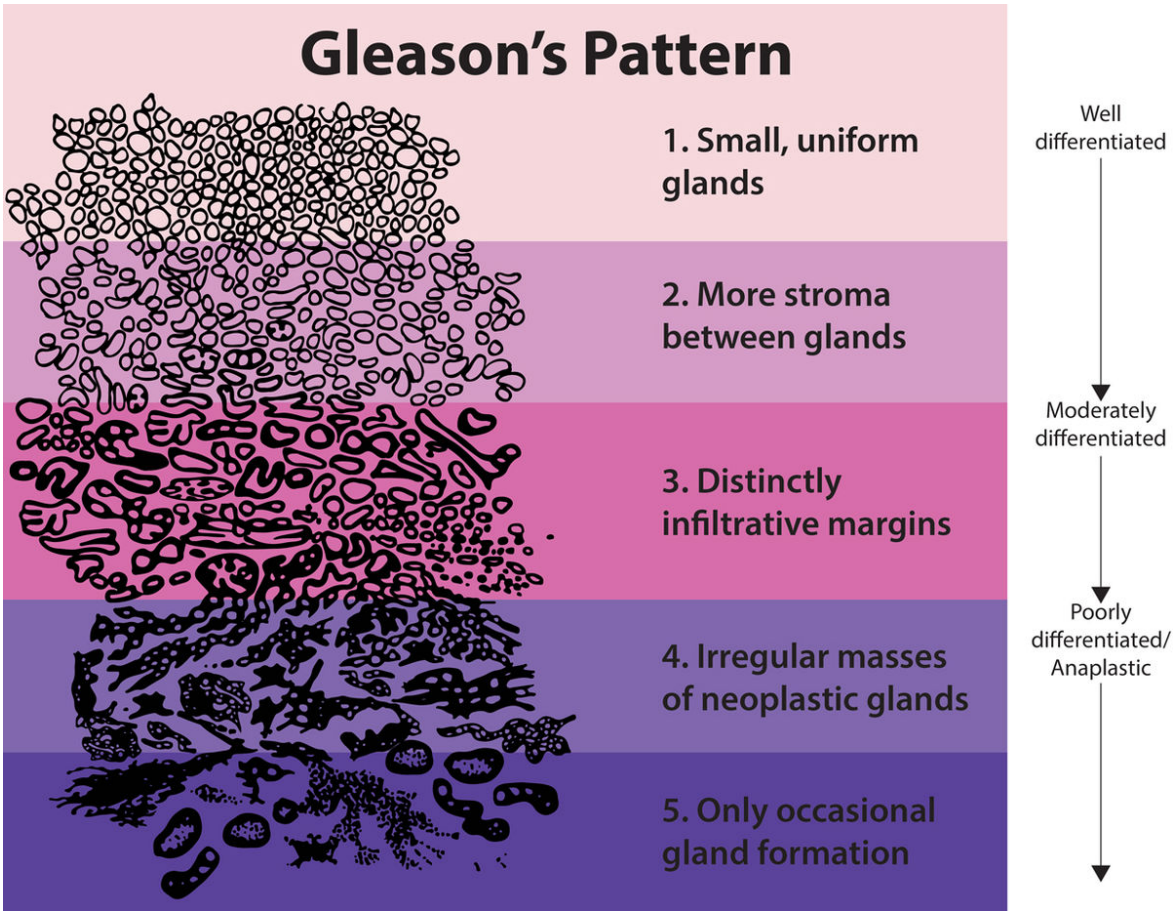

Figure A.1: Gleason grading diagram. Reprinted from [140].

it still leaves some "noisy prediction" in between of the gland structures. We argue that the reason behind this phenomenon comes from the fact that the softmax layer requires the prediction to compete with different classes at each pixel, which enforces the model to "think" in a pixel-wise level. On the other hand, our proposed R-CNN method is a region based method wherein the label prediction is based on a super-pixel group, the "instance." For each instance, the model will give one label prediction. Decoupling the classification and segmentation problem therefore eliminates the "noisy prediction" that appears in the Multi-scale U-Net.

Our initial implementation of R-CNN did not perform as expected as it performed similar to the Multi-scale U-Net. One key observation we had after carefully reviewing the results

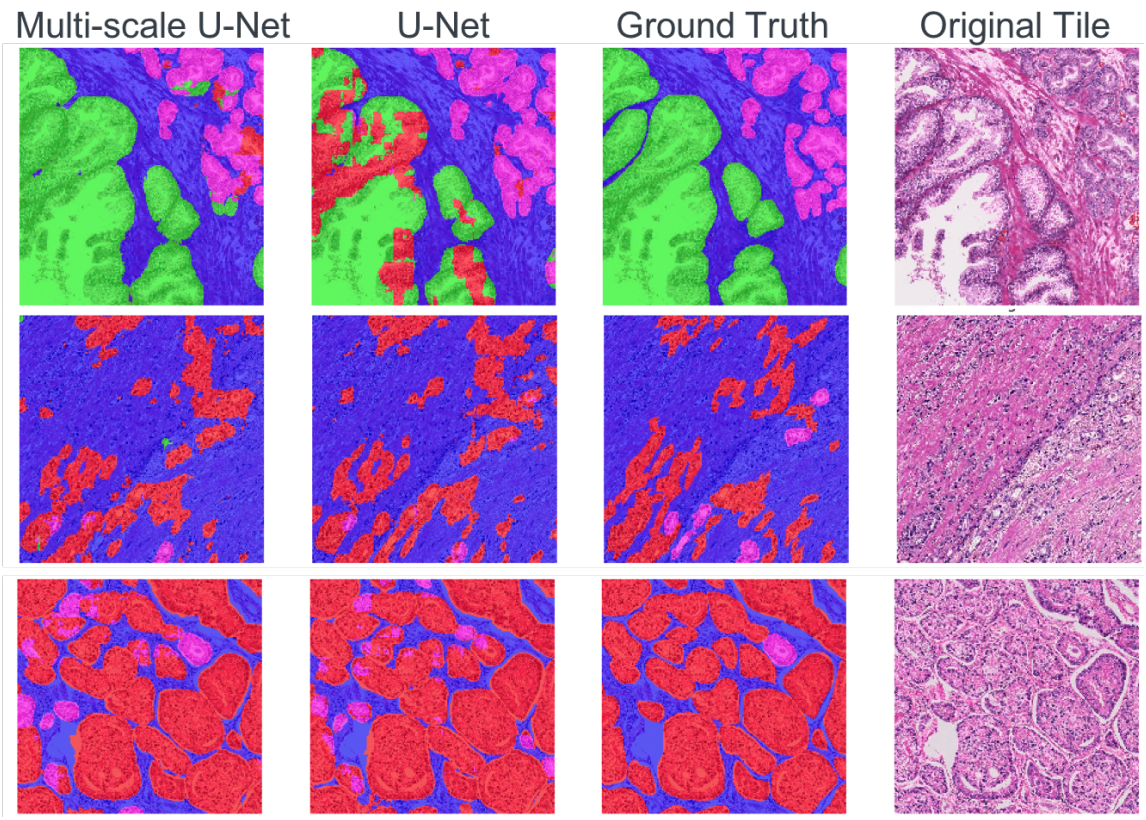

Figure A.2: Results of Multi-Scale U-Net

was that the Path R-CNN model had difficulty distinguishing between stroma and epithelial cells (especially high-grade cancer regions, see Figure 2.5 Column 4). Therefore, we first added another individual network to distinguish the presence of epithelial cells prior to the R-CNN. As stated in Section 2.3.2.3 ENH Effect, we empirically found that explicitly adding this network performed much better than tuning the "detection threshold" inside the GNH. To reduce the computational overhead, we moved this individual network after the ResNet backbone, resulting a more efficient Epithelial Network Head (ENH).

As the result from Path R-CNN w/o the ENH is nearly same as the Multi-scale U-Net results, a natural question to ask is: can the Multi-scale U-Net combined with the ENH prediction suppress false positives in non-epithelial cell regions? We did an additional experiment by combining the ENH with the Multi-scale U-Net. As listed in Table A.1, it actually improves the performance by 1.2% in mIOU, which is not as large as the Path

R-CNN. In Figure A.3, we show the prediction results of the Multi-scale U-Net and the Path R-CNN with and without EPH. We argue that while the R-CNN method suffers from distinguishing the stroma from the epithelial cells, the Multi-scale U-Net performs very well on stroma classification. Thus adding the EPH explicitly does not improve performance much.

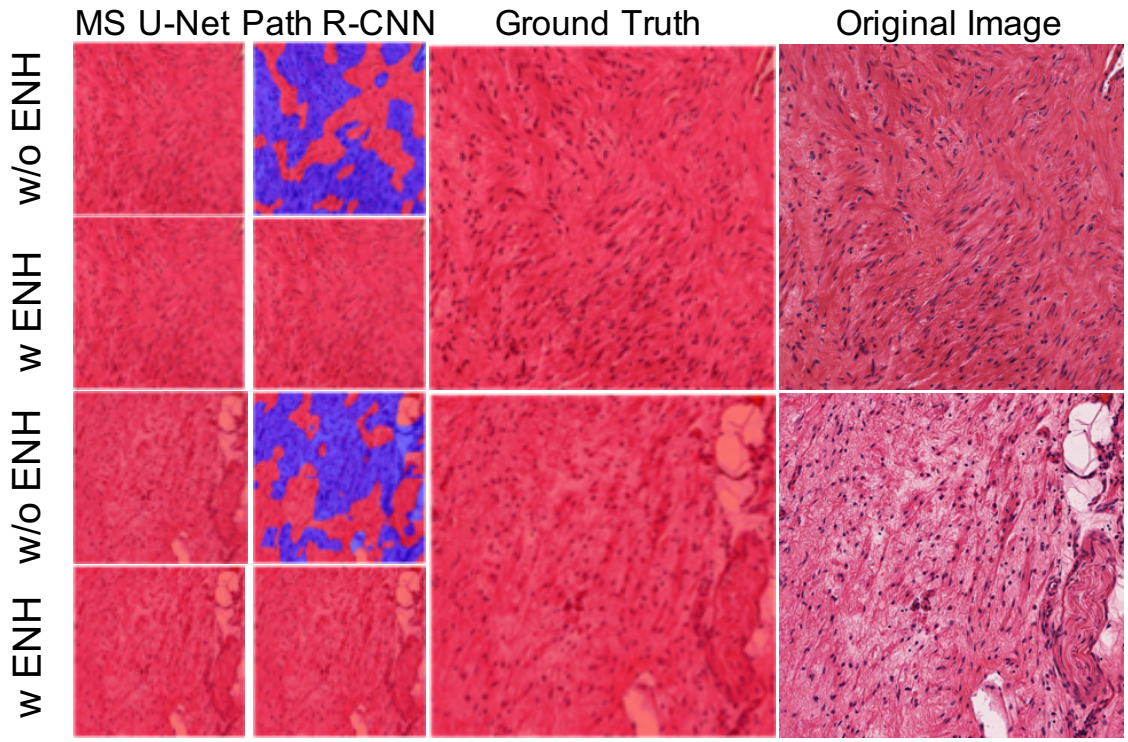

Figure A.3: Multi Scale U-Net model prediction with and without ENH compared with Path R-CNN.

Table A.1: Multi-scale U-Net performance with ENH

	J_{BG}	J_{BN}	J_{LG}	J_{HG}	$mIOU$
Multi-Scale U-Net	82.42%	72.13%	58.70%	78.38%	72.91%
Multi-Scale U-Net w ENH	NA%	NA%	NA%	NA%	NA%

A.3 Effect of Transfer Learning

We empirically found that the pre-training network did not necessarily give more accurate results. However, it did accelerate the convergence rate during model training. As shown in Figure A.4, the training loss without transfer learning decreases slowly compared with transfer learning. The loss without transfer learning decreases to the same "stopping level" at epoch 90 compared to epoch 70 with transfer learning. [47] provided a possible explanation on why transfer learning only helps with model convergence and not model performance. Though different initialization weights will lead to different "local minimums" in the end, these "local minimums" can join to form macroscopic basins (known as Hopfield memories in the neural network community), which results in similar performance. The detailed discussion is beyond the scope of our thesis, but we refer interested readers to [47] for more information.

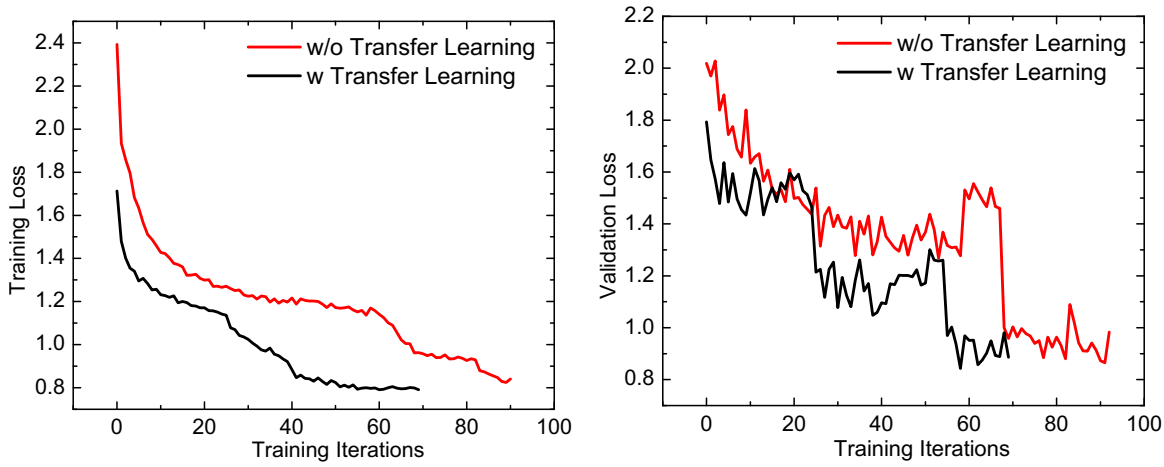

Figure A.4: Impact of Transfer Learning on Model Convergence

APPENDIX B

Appendix for Chapter 3

B.1 Network Architecture

We list the detailed architecture we used to compare Good GAN and Bad GAN on MNIST, SVHN, and CIFAR10 datasets in Table B.1, Table B.2 and Table B.3 respectively.

Table B.1: MNIST

Generator G	**Classifier C**	**Discriminator D (Good GAN only)**
Input Label y, Noise z	Input 28×28 Gray Image	Input 28×28 Gray Image, Label y
MLP 500 units, softplus, batch norm	MLP 1000 units, lRelu, Gaussian noise, weight norm	MLP 1000 units, lRelu, Gaussian noise, weight norm
MLP 500 units, softplus, batch norm	MLP 500 units, lRelu, Gaussian noise, weight norm	MLP 500 units, lRelu, Gaussian noise, weight norm
MLP 500 units, softplus, batch norm	MLP 250 units, lRelu, Gaussian noise, weight norm	MLP 250 units, lRelu, Gaussian noise, weight norm
MLP 500 units, softplus, batch norm	MLP 250 units, lRelu, Gaussian noise, weight norm	MLP 250 units, lRelu, Gaussian noise, weight norm
	MLP 10 units, softmax, Gaussian noise, weight norm	MLP 12 units, sigmoid, Gaussian noise, weight norm

B.2 Batch Size Effect in Bad GAN

Figure B.2 shows the classification accuracy under different batch sizes for Bad GAN during the first 400 epochs of training. As can be seen, the model performance is very sensitive to batch size. Figure B.1 shows the generated images of Good GAN under different batch sizes. With small batch size, Good GAN is not able to generate good image-label pairs.

Table B.2: SVHN

Generator G	**Classifier C**	**Discriminator D (Good GAN only)**
Input Label y, Noise z	Input 32×32 Colored Image	Input 32×32 Colored Image, Label y
MLP 8192 units, Relu, batch norm Reshape $512 \times 4 \times 4$ 5×5 deconv. 256. stride 2, Relu, batch norm	Gaussian noise, 0.2 dropout 3×3 conv. 64. lRelu, weight norm 3×3 conv. 64. lRelu, weight norm 3×3 conv. 64. lRelu, stride 2, weight norm 0.5 dropout	0.2 dropout 3×3 conv. 32. lRelu, weight norm 3×3 conv. 32. lRelu, stride 2, weight norm 0.2 dropout
5×5 deconv. 128. stride 2, Relu, batch norm	3×3 conv. 128. lRelu, weight norm 3×3 conv. 128. lRelu, weight norm 3×3 conv. 128. lRelu, stride 2, weight norm 0.5 dropout	3×3 conv. 64. lRelu, weight norm 3×3 conv. 64. lRelu, stride 2, weight norm 0.2 dropout
5×5 deconv. 3. stride 2, sigmoid, weight norm	3×3 conv. 128. lRelu, weight norm 3×3 conv. 128. lRelu, weight norm 3×3 conv. 128. lRelu, weight norm Global pool MLP 10 units, softmax, weight norm	3×3 conv. 128. lRelu, weight norm 3×3 conv. 128. lRelu, weight norm Global pool MLP 1 unit, sigmoid, weight norm

Table B.3: CIFAR10

Generator G	**Classifier C**	**Discriminator D (Good GAN only)**
Input Label y, Noise z	Input 32×32 Colored Image	Input 32×32 Colored Image, Label y
MLP 8192 units, Relu, batch norm Reshape $512 \times 4 \times 4$ 5×5 deconv. 256. stride 2, Relu, batch norm	Gaussian noise, 0.2 dropout 3×3 conv. 96. lRelu, weight norm 3×3 conv. 96. lRelu, weight norm 3×3 conv. 96. lRelu, stride 2, weight norm 0.5 dropout	0.2 dropout 3×3 conv. 32. lRelu, weight norm 3×3 conv. 32. lRelu, stride 2, weight norm 0.2 dropout
5×5 deconv. 128. stride 2, Relu, batch norm	3×3 conv. 192. lRelu, weight norm 3×3 conv. 192. lRelu, weight norm 3×3 conv. 128. lRelu, stride 2, weight norm 0.5 dropout	3×3 conv. 64. lRelu, weight norm 3×3 conv. 64. lRelu, stride 2, weight norm 0.2 dropout
5×5 deconv. 3. stride 2, sigmoid, weight norm	3×3 conv. 192. lRelu, weight norm 3×3 conv. 192. lRelu, weight ntheirorm 3×3 conv. 192. lRelu, weight norm Global pool MLP 10 units, softmax, weight norm	3×3 conv. 128. lRelu, weight norm 3×3 conv. 128. lRelu, weight norm Global pool MLP 1 unit, sigmoid, weight norm

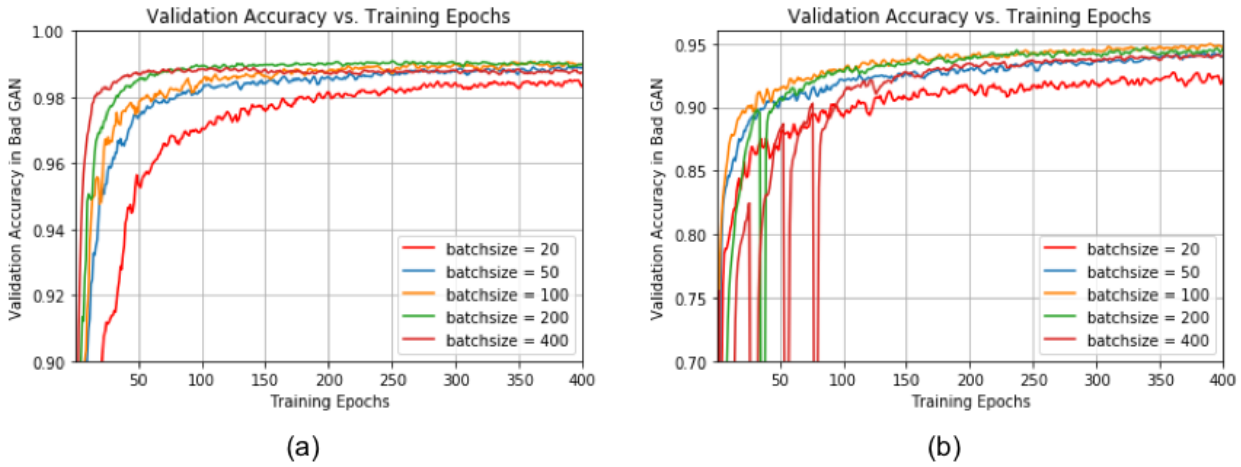

Figure B.1: Batch size effect in Bad GAN. The classification accuracy over the initial 400 training epochs under different batch size. (a) The experiments are performed on MNIST dataset, using 100 labeled data. (b) The experiments are performed on SVHN dataset, using 1000 labeled data.

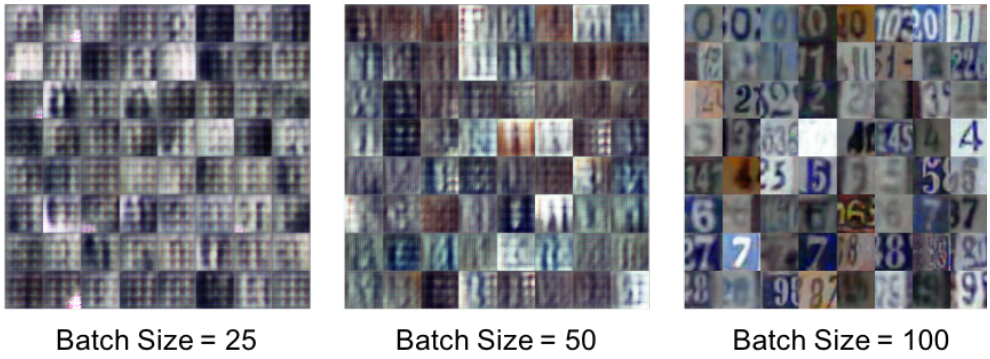

Figure B.2: Batch size effect in Good GAN. With small batch size, Good GAN is not able to generate good image-label pairs. Experiments are performed on SVHN with $n = 1000$. All the images are generated at epoch = 200 when we start to use the generated image to train.

APPENDIX C

Appendix for Chapter 4

C.1 Loss Function of the Classifier

Softmax layer is over-parameterized, therefore we can still model C with K neurons at the output layer. To represent $K+1$ classes, the loss function should be modified as detailed below.

First let us rewrite the four components of C's objective function:

$$\begin{aligned} L_{C_1} &= -\mathbb{E}_{x,y\sim p_l(x,y)}[\log(p_C(y|x, y \leqslant K)] \qquad L_{C_2} = -\mathbb{E}_{x,y\sim p_{gG}(x,y)}[\log(p_C(y|x, y \leqslant K)] \\ L_{C_3} &= -\mathbb{E}_{x\sim p_u(x)}[\log(1 - p_C(y = K+1|x)] \quad L_{C_4} = -\mathbb{E}_{x\sim p_{bG}(x)}[\log(p_C(y = K+1|x)] \end{aligned} \tag{C.1}$$

Suppose $\{l_1(x), l_2(x), l_3(x), \cdots, l_K(x), l_{K+1}(x)\}$ represents the logits before the softmax-layer for input x, by using the fact that softmax is over-parameterized, we can fix the logit $l_{K+1}(x) = 0 \ \forall x$ for the bG generated images and the output of the softmax remains the same. Hence, we can reformulate the above four components as

$$\begin{aligned} L_{C_1} &= -\mathbb{E}_{x,y\sim p_l(x,y)}[-l_y + \log(\sum_{i=1}^{K} \exp l_i)] \\ L_{C_2} &= -\mathbb{E}_{x,y\sim p_{gG}(x,y)}[-l_y + \log(\sum_{i=1}^{K} \exp l_i)] \\ L_{C_3} &= -\mathbb{E}_{x\sim p_u(x)}[-\log(\sum_{i=1}^{K} \exp l_i) + \log(1 + \sum_{i=1}^{K} \exp l_i)] \\ L_{C_4} &= -\mathbb{E}_{x\sim p_{bG}(x)}[\log(1 + \sum_{i=1}^{K} \exp l_i)] \end{aligned} \tag{C.2}$$

Define the log sum exponent function as $\mathrm{LSE}(\boldsymbol{x}) = \log(\sum_j \exp x_j)$ and softplus function

as $\text{softplus}(x) = \log(1 + \exp x)$, the losses can be further simplified as

$$\begin{aligned} L_{C_1} &= -\mathbb{E}_{x,y\sim p_l(x,y)}[-l_y + \text{LSE}(\boldsymbol{l})] \\ L_{C_2} &= -\mathbb{E}_{x,y\sim p_{gG}(x,y)}[-l_y + \text{LSE}(\boldsymbol{l})] \\ L_{C_3} &= -\mathbb{E}_{x\sim p_u(x)}[-\text{LSE}(\boldsymbol{l}) + \text{softplus}(\text{LSE}(\boldsymbol{l}))] \\ L_{C_4} &= -\mathbb{E}_{x\sim p_{bG}(x)}[\text{softplus}(\text{LSE}(\boldsymbol{l}))] \end{aligned} \tag{C.3}$$

which are used in our code implementation.

C.2 How does bG work?

In order to get a more intuitive understanding of why a complement generator could boost SSL performance, we conduct analysis experiments based on 2D synthetic data. As shown in the Figure C.1, labeled and unlabeled data are denoted by a dark-colored triangle and a light-colored circle respectively, and two classes are indicated by different colors. We add fake data points (denoted by yellow triangles) which lie between the data manifolds of two classes in Figure C.1(c). The exact same model and parameters are then used to train binary-classification models with two groups of data points. We visualize decision boundaries of these two classification models. As expected, the decision boundary in Figure C.1(d) always lies in the fake data area outside data manifolds, which in turn improved generalized ability of model.

C.3 Detailed Theoretical Analysis

Lemma 1 *For any fixed C and G, the optimal D of the game defined by the loss function (1) - (5) is*

$$D^*_{C,gG,bG}(x,y) = \frac{p_l(x,y)}{p_l(x,y) + p_{\frac{1}{2}}(x,y)}, \tag{C.4}$$

where $p_{\frac{1}{2}}(x,y) = \frac{1}{2}p_{gG}(x,y) + \frac{1}{2}p_C(x,y|y \leqslant K)$.

Proof: This follows from Proposition 1 of [10] directly.

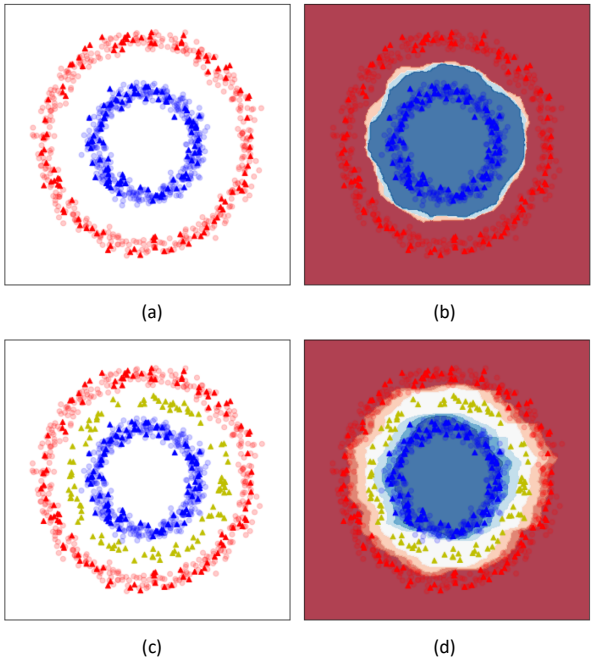

Figure C.1: Effectiveness of bG on synthetic data. (a) data points without fake data generated by bG; (b) decision boundary without bG; (c)data points with fake data generated by bG; (d) decision boundary with bG

Theorem 2 *The global minimum of* $V(C, gG, bG)$ *is achieved only when* $p_l(x, y) = p_{gG}(x, y) = p_C(x, y|y \leqslant K)$.

Proof:

Given $D^*_{C,gG,bG}$, we can reformulate our value function as

$$V(C, gG, bG) = -\log 4 + 2JSD(p_l(x, y), p_{\frac{1}{2}}(x, y)) + L_{C1} + L_{C2} + L_{C3} + L_{C4}. \tag{C.5}$$

We first focus on the term with respect to $p_C(x, y|y \leqslant K)$, denoted the corresponding loss as $\tilde{V}(C|y \leqslant K)$, we have

$$\begin{aligned}\tilde{V}(C|y \leqslant K) \propto & 2JSD(p_l(x, y), p_{\frac{1}{2}}(x, y)) - \mathbb{E}_{x,y\sim p_l(x,y)}[\log(p_C(y|x, y \leqslant K)] \\ & - \mathbb{E}_{x,y\sim p_{gG}(x,y)}[\log(p_C(y|x, y \leqslant K)] \\ \propto & 2JSD(p_l(x, y), p_{\frac{1}{2}}(x, y)) + KL(p_\beta(x, y)||p_C(y|x, y \leqslant K),\end{aligned} \tag{C.6}$$

where $p_\beta(x, y) = \beta p_l(x, y) + (1 - \beta)p_{gG}(x, y)$ and $\beta/(1 - \beta)$ is the ratio of data we feed into classifier between true labeled data and data pairs from good generator. Therefore the global minimum can only be achieved when

$$\begin{aligned}p_l(x, y) &= \frac{1}{2}p_{gG}(x, y) + \frac{1}{2}p_C(x, y|y \leqslant K) \\ p_C(x, y|y \leqslant K) &= \beta p_l(x, y) + (1 - \beta)p_{gG}(x, y),\end{aligned} \tag{C.7}$$

and it is obtained when $p_l(x, y) = p_{gG}(x, y) = p_C(x, y|y \leqslant K)$.

Corollary 2.1 *The optimal classifier* C *will have* $p_C(y = K + 1|x \sim p_u(x)) = 0$ *and* $p_C(y = K + 1|x \sim p_{bG}(x)) = 1$.

Proof: Because $p_C(y = K + 1|x)$ and $p_C(y|x, y \leqslant K)$ are independent, we can consider them separately. The term related to $p_C(y = K + 1|x)$ in loss function is

$$L_{C_3} + L_{C_4} = -\mathbb{E}_{x\sim p_u(x)}[\log(1 - p_C(y = K + 1|x)] - \mathbb{E}_{x\sim p_{bG}(x)}[\log(p_C(y = K + 1|x)], \tag{C.8}$$

which achieves its minimal 0 when $p_C(y = K + 1|x \sim p_u(x)) = 0$ and $p_C(y = K + 1|x \sim p_{bG}(x)) = 1$.

Corollary 2.2 *If applying the iterative procedure described in (9) and (10),*

$$\mathrm{KL}(p(y_l|x)||p_{\theta_{s+1}}(y_l|x, y \leqslant K)) \leqslant \mathrm{KL}(p(y_l|x)||p_{\theta_s}(y_l|x, y \leqslant K)) \tag{C.9}$$

Proof: Define

$$J(\theta, p(Z|x)) = \mathrm{KL}(p(y_l|x)p(Z|x)||p_\theta(y_l, Z|x, y \leqslant K)), \tag{C.10}$$

and

$$J(\theta) = \mathrm{KL}(p(y_l|x)||p_\theta(y_l|x, y \leqslant K)). \tag{C.11}$$

Then we have

$$J(\theta_{s+1}) \leqslant J(\theta_{s+1}, p_{\theta_s}(Z|x)) \leqslant J(\theta_s, p_{\theta_s}(Z|x)) = J(\theta_s). \tag{C.12}$$

C.4 Datasets

We apply UGAN on the widely adopted MNIST [77], SVHN [99], and CIFAR10 [69] datasets. MNIST consists of 50,000 training samples, 10,000 validation samples, and 10,000 testing samples of handwritten digits of size 28×28. SVHN consists of 73,257 training samples and 26,032 testing samples. Each sample is a colored image of size 32×32, containing a sequence of digits with various backgrounds. CIFAR10 consists of colored images distributed across 10 general classes – *airplane*, *automobile*, *bird*, *cat*, *deer*, *dog*, *frog*, *horse*, *ship* and *truck*. It contains 50,000 training samples and 10,000 testing samples of size 32×32. Following [19], we reserve 5,000 training samples from SVHN and CIFAR10 for validation if needed in our experiments.

C.5 Network Architecture

We list the detailed architecture we used to construct UGAN in Table C.1, Table C.2 and Table C.3 respectively. To re-implement Triple-GAN and Bad GAN, we also use the same architecture of the corresponding parts for fair comparison. Note that in Bad GAN, the discriminator has two roles: to classify the real data into the right class and to distinguish the real samples from the fake samples. For clarity, we refer to Bad GAN's D as C in the table, while D is a conditional network that presents in Triple-GAN and UGAN.

Table C.1: MNIST

bG	gG	C	D
$z \sim p(z)$	$y \sim p(y)$, $z \sim p(z)$	$x \sim p_{\{l,u,gG,bG\}}(x)$	$(x, y) \sim p_{\{l,gG,C\}}(x, y)$
MLP 500 units, softplus, batch norm MLP 500 units, softplus, batch norm MLP 500 units, softplus, batch norm		MLP 1000 units, lRelu, Gaussian noise, weight norm MLP 500 units, lRelu, Gaussian noise, weight norm MLP 250 units, lRelu, Gaussian noise, weight norm MLP 250 units, lRelu, Gaussian noise, weight norm MLP 250 units, lRelu, Gaussian noise, weight norm MLP 10 units, softmax, Gaussian noise, weight norm	MLP 1000 units, lRelu, Gaussian noise, weight norm MLP 500 units, lRelu, Gaussian noise, weight norm MLP 250 units, lRelu, Gaussian noise, weight norm MLP 250 units, lRelu, Gaussian noise, weight norm MLP 250 units, lRelu, Gaussian noise, weight norm MLP 12 units, sigmoid, Gaussian noise, weight norm

C.6 Results of Varying Amount of Labeled Data

We perform our experiments on setups with 20, 50, 100, and 200 labeled examples in MNIST, 500, 1000, and 2000 labeled examples in SVHN, and 1000, 2000, 400, 8000 examples in CIFAR10. Table C.4 $\sim$ C.5 show the results of the experiemts on SVHN, and CIFAR10 respectively. We find that our UGAN constantly outperforms Triple-GAN and Bad GAN across a wide range of labled data.

C.7 Importance of Selected Labeled Data

One interesting observation is that the selection of labeled data plays a crucial role for training Triple-GAN, Bad GAN and UGAN in the low labeled data scenario. For most cases, the labeled data used for the training in our experiments are randomly selected stratified samples, except for the MNIST-20 case. In this case, we found selecting representative

Table C.2: SVHN

bG	gG	C	D
$z \sim p(z)$	$y \sim p(y)$, $z \sim p(z)$	$x \sim p_{\{l,u,gG,bG\}}(x)$	$(x, y) \sim p_{\{l,gG,C\}}(x, y)$
MLP 8192 units, Relu, batch norm Reshape $512 \times 4 \times 4$ 5×5 deconv. 256. stride 2, Relu, batch norm		Gaussian noise, 0.2 dropout 3×3 conv. 64. lRelu, weight norm 3×3 conv. 64. lRelu, weight norm 3×3 conv. 64. lRelu, stride 2, weight norm 0.5 dropout	0.2 dropout 3×3 conv. 32. lRelu, weight norm 3×3 conv. 32. lRelu, stride 2, weight norm 0.2 dropout
5×5 deconv. 128. stride 2, Relu, batch norm		3×3 conv. 128. lRelu, weight norm 3×3 conv. 128. lRelu, weight norm 3×3 conv. 128. lRelu, stride 2, weight norm 0.5 dropout	3×3 conv. 64. lRelu, weight norm 3×3 conv. 64. lRelu, stride 2, weight norm 0.2 dropout
5×5 deconv. 3. stride 2, sigmoid, weight norm		3×3 conv. 128. lRelu, weight norm 3×3 conv. 128. lRelu, weight norm 3×3 conv. 128. lRelu, weight norm Global pool MLP 10 units, softmax, weight norm	3×3 conv. 128. lRelu, weight norm 3×3 conv. 128. lRelu, weight norm Global pool MLP 1 unit, sigmoid, weight norm

Table C.3: CIFAR10

bG	gG	C	D
$z \sim p(z)$	$y \sim p(y)$, $z \sim p(z)$	$x \sim p_{\{l,u,gG,bG\}}(x)$	$(x, y) \sim p_{\{l,gG,C\}}(x, y)$
MLP 8192 units, Relu, batch norm Reshape $512 \times 4 \times 4$ 5×5 deconv. 256. stride 2, Relu, batch norm		Gaussian noise, 0.2 dropout 3×3 conv. 96. lRelu, weight norm 3×3 conv. 96. lRelu, weight norm 3×3 conv. 96. lRelu, stride 2, weight norm 0.5 dropout	0.2 dropout 3×3 conv. 32. lRelu, weight norm 3×3 conv. 32. lRelu, stride 2, weight norm 0.2 dropout
5×5 deconv. 192. stride 2, Relu, batch norm		3×3 conv. 192. lRelu, weight norm 3×3 conv. 192. lRelu, weight norm 3×3 conv. 192. lRelu, stride 2, weight norm 0.5 dropout	3×3 conv. 64. lRelu, weight norm 3×3 conv. 64. lRelu, stride 2, weight norm 0.2 dropout
5×5 deconv. 3. stride 2, sigmoid, weight norm		3×3 conv. 192. lRelu, weight norm 3×3 conv. 192. lRelu, weight norm 3×3 conv. 192. lRelu, weight norm Global pool MLP 10 units, softmax, weight norm	3×3 conv. 192. lRelu, weight norm 3×3 conv. 192. lRelu, weight norm Global pool MLP 1 unit, sigmoid, weight norm

labeled data to train is the key to achieving good performance. The reported accuracy in Table 2 is averaged over 10 runs where we manually selected different representative labeled

Table C.4: Test accuracy on semi-supervised SVHN. Results are averaged over 10 runs.

Model	Test accuracy for a given number of labeled samples		
	500	1000	2000
Bad GAN [5]	-	$95.75 \pm 0.03\%$	-
Triple-GAN [4]	-	$94.23 \pm 0.17\%$	-
Bad GAN (ours)	$94.21 \pm 0.45\%$	$95.32 \pm 0.07\%$	$95.47 \pm 0.39\%$
Triple-GAN (ours)	$94.67 \pm 0.12\%$	$95.30 \pm 0.38\%$	$95.37 \pm 0.09\%$
UGAN	$\mathbf{95.53 \pm 0.13\%}$	$\mathbf{96.49 \pm 0.09\%}$	$\mathbf{96.51 \pm 0.05\%}$

Table C.5: Test accuracy on semi-supervised CIFAR10. Results are averaged over 10 runs.

Model	Test accuracy for a given number of labeled samples			
	1000	2000	4000	8000
Bad GAN [5]	-	-	$85.59 \pm 0.03\%$	-
Triple-GAN [4]	-	-	$83.01 \pm 0.36\%$	-
Bad GAN (ours)	$77.58 \pm 0.17\%$	$81.36 \pm 0.08\%$	$82.89 \pm 0.13\%$	$85.47 \pm 0.10\%$
Triple-GAN (ours)	$81.08 \pm 0.57\%$	$81.79 \pm 0.37\%$	$82.82 \pm 0.41\%$	$85.37 \pm 0.18\%$
UGAN	$\mathbf{82.34 \pm 0.17\%}$	$\mathbf{83.88 \pm 0.13\%}$	$\mathbf{85.66 \pm 0.06\%}$	$\mathbf{86.58 \pm 0.09\%}$

data in a stratified way. Figure C.2(a) shows a single run that UGAN uses randomly selected labeled data and does not achieve good results, while Figure C.2(b) shows another run that is able to achieve higher accuracy. The failure of the first run is due to the initial selections for digit 4 being similar to 9, causing the generator to generate many 9s when conditioned on label 4. The generator also generates low-quality images. We also report that with a random selection of 20 labeled data, Triple-GAN is able to achieve $76.78 \pm 6.47\%$ accuracy over 3 runs, Bad GAN is achieving $68.12 \pm 0.60\%$ over 10 runs, and UGAN is able to achieve $89.35 \pm 7.61\%$ accuracy over 3 runs. As can be seen, in both cases Triple-GAN outperfoms Bad GAN, while UGAN outperforms both of them, revealing that UGAN is least sensitive to

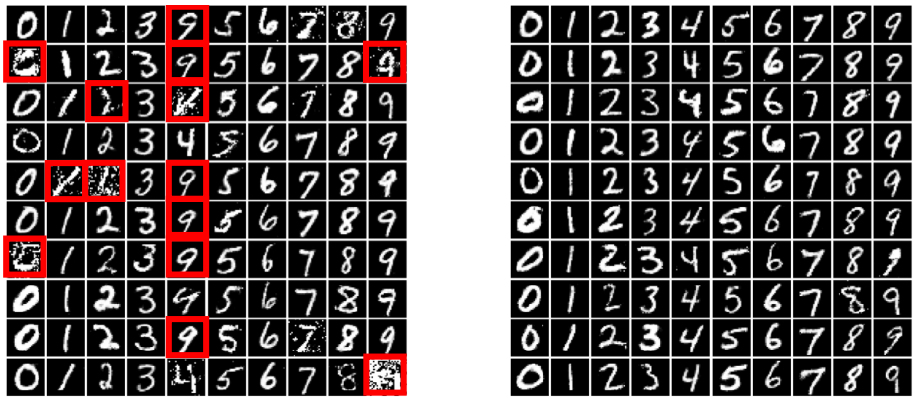

Figure C.2: Two-runs of UGAN model on MNIST dataset. (a) A single run where we randomly select 20 labeled data. gG generates a lot of wrong images conditioned on the label, resulting in bad performance of C. (b) Another run where we manually select 20 representative labeled examples. This time gG is able to generate correct images, and C achieves good classification performance.

the amounts of labeled data. The importance of selected labeled data is not surprising and is related to active learning, a potential future work could be extending UGAN for active learning.

C.8 Generator Evolution

By iteratively updating D, gG, C, and bG using gradient descent, UGAN is able to obtain a good generator and a bad generator simultaneously. To illustrate this, Figure C.3 shows an evolution of both gG and bG generated samples throughout the training on MNIST, SVHN, and CIFAR10. As the training progresses, gG generated samples become clearer and semantic meaningful; bG generated samples are more close to data manifold but semantically meaningless.

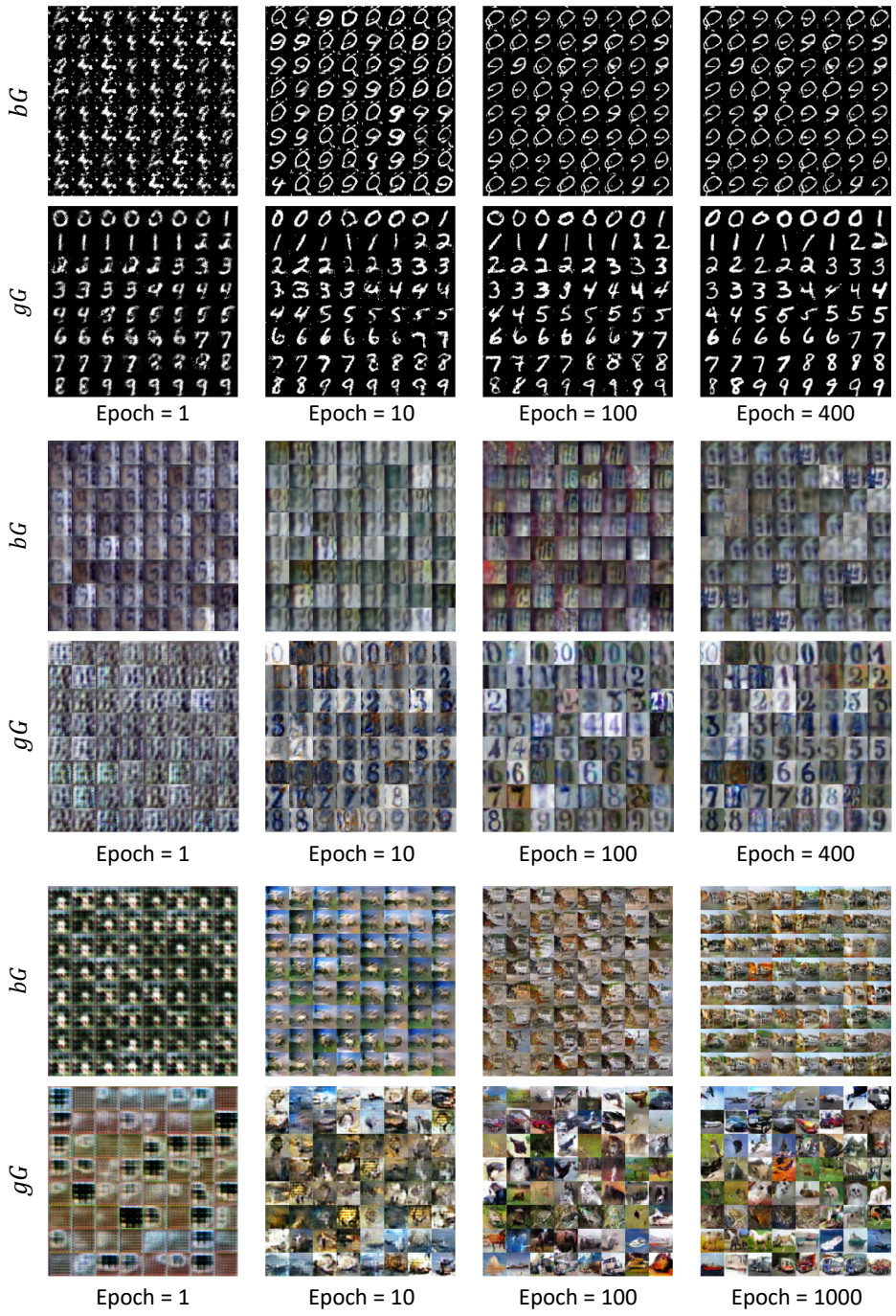

Figure C.3: gG and bG evolution. Generated images from both bG and gG throughout training are shown. UGAN are trained on MNIST (upper), SVHN (middle), and CIFAR10 (lower). Through training, UGAN is able to obtain a good generator and a bad generator simultaneously.

C.9 Hyper-parameters Sensitivity Analysis

GAN-based methods do require hyper-parameter tuning on a relatively large dataset. Nevertheless, we found that the hyper-parameters used in Triple-GAN and Bad GAN are good starting points for UGAN. In fact, aside from batch size, we use the same hyper-parameters across all three datasets, and consistently achieved good results. We perform hyper-parameter sensitivity analysis, along with some network architecture effects, which are summarized in this Section. Table C.6 summarizes the initial learning rate effect on final model performance. The experiments are done on MNIST $n = 100$. Despite the differences of the training loss in the initial stage, the final results are not significantly different after training 400 epochs, indicating the algorithm is not sensitive to learning rate, which is expected when using the Adam optimizer. However, as we mention in the chapter, UGAN is sensitive to batch size. Besides the experimental settings in the chapter, we also apply different batch sizes in SVHN $n = 1000$. UGAN fails to perform well when using 25 and 50 in gG since it cannot generate reliable image-label pairs. Using 100 in bG, UGAN achieves 96.31% accuracy, around a 0.2% drop in accuracy. The results indicate the batch noise benefits the bG, while it hurts the gG capability to model the true data distribution. For the model architecture, we find that the weight-norm layer is important to ease GAN's training. We also use a smaller architecture of C with filter size {32, 64, 96} and get 96.27% on SVHN n = 4000. No significant drop compared to our reported results indicates that UGAN is robust to model architecture in a range.

Table C.6: Initial Learning Rate Effect on Model Performance.

Learning Rate	$lr = 1e-2$	$lr = 1e-3$	$lr = 5e-4$	$lr = 3e-4$
Accuracy	99.13%	99.18%	99.24%	99.18%

C.10 Good and Bad Samples Effectiveness

As mentioned in Section 4.4, we also observe a similar three phases training process in MNIST and CIFAR10. Figure C.4(a) and (b) show the comparison among Triple-GAN, Bad GAN, and UGAN on MNIST and CIFAR10 respectively. The experiments are done under MNIST $n = 100$ and SVHN $n = 1000$.

For the number of labeled data effect, we don't find a similar transition on SVHN and CIFAR10 as in Fig. 3(b). Instead, we find a gradual change of the learning curve under different amounts of labeled data. We also have tried to push the number of labeled data even lower (i.e., $n < 500$ in SVHN and $n < 1000$ in CIFAR10), but UGAN fails to generate good image-label pairs. One possible explanation is that when we use too few labeled data, gG fails to model the conditional distribution due to the complexity of SVHN and CIFAR10.

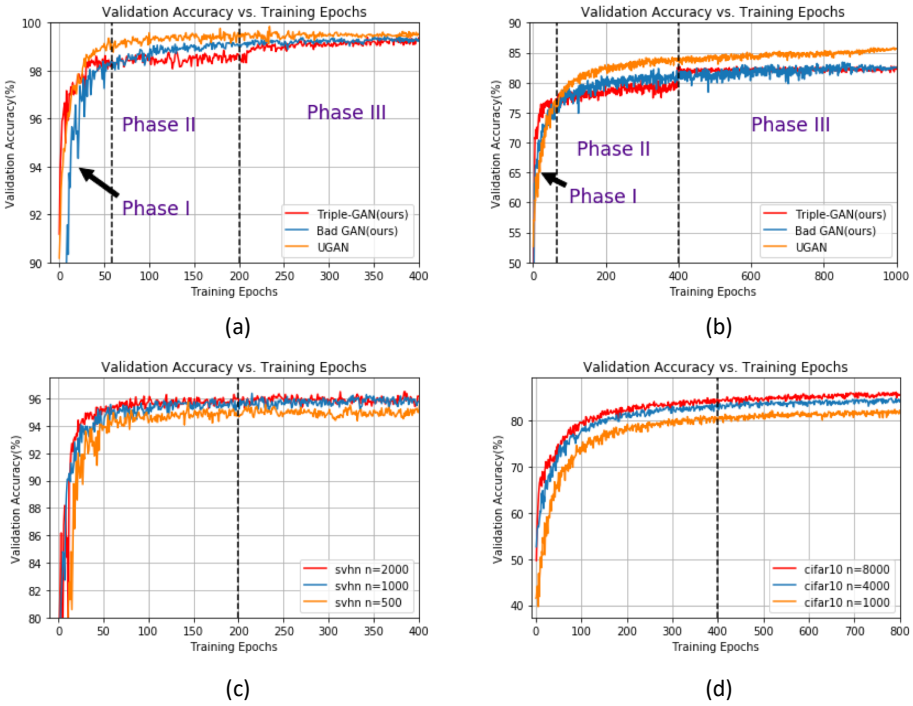

Figure C.4: Comparison of Triple-GAN, Bad GAN, and UGAN on (a) MNIST $n = 100$ and (b) SVHN $n = 1000$. Similar three-phase training processes have been observed in both cases. UGAN Validation Accuracy vs. Training Epochs under various amount of labeled data on (c) SVHN and (d) CIFAR10. We don't find a similar transition on SVHN and CIFAR10 as in Fig. 3(b). The vertical dot line in (c) and (d) denotes the epoch when we start to use gG generated image-label pairs to train C.

APPENDIX D

Appendix for Chapter 5

D.1 Network Architectures

We list the detailed network architecture we used as G_n, S_n, and D_n in Table D.1. Note that the discriminator D_n consists of a multi-scale structure. Each scale has the same network architecture. For simplicity, we only list one-level network layout in the table.

Table D.1: Network Architecture for G_n, S_n and D_n.

G_n	S_n	D_n
$(\tilde{x}^{\uparrow}_{n+1}, y_n, z_n)$	$(x_n, \tilde{y}^{\uparrow}_{n+1}, z_n)$	$(x_n, y_n), (\tilde{x}_{G_n}, y_n), (x_n, \tilde{y}_{S_n})$
ReflectionPadding, Conv 7×7	Conv 3×3, BatchNorm, ReLU	
Instance-Norm, ReLU	Downsampling:	
Downsampling:	(DenseBlock (BatchNorm, ReLU),	Single-Layer Discriminator:
(Conv 3×3, Instance-Norm, ReLU)*2	Transition Down)*3	Conv 4×4, Instance-Norm, lReLU
Resnet Blocks:	Bottleneck: DenseBlock	Conv 4×4, Instance-Norm, lReLU
ResnetBlocks (Instance-Norm, ReLU) * 2	Upsampling:	Conv 4×4, Instance-Norm, lReLU
Upsampling:	(DenseBlock (BatchNorm, ReLU),	Conv 4×4, Instance-Norm, lReLU
(Deconv 3×3, Instance-Norm, ReLU)*2	Transition Up)*3	Conv 4×4, Instance-Norm, Sigmoid
ReflectionPadding, Conv 7×7	Conv 3×3, ReLU	
Instance-Norm, Tanh	Softmax	

D.2 Theoretical Analysis

We now provide theoretical analysis of G_n, S_n, and D_n at each scale. First we can show that the optimal D_n balances between the true data distribution and the mixture distribution

defined by G_n and S_n, as summarized in Lemma 3.

Lemma 3. *For any fixed S_n and G_n, the optimal D_n of the game defined by loss functions (5.2) and (5.6) is*

$$D^*_{n(G_n,S_n)}(x,y) = \frac{p(x,y)}{p(x,y) + p_{\frac{1}{2}}(x,y)}, \tag{D.1}$$

where $p_{\frac{1}{2}}(x,y) = \frac{1}{2}p_{G_n}(x,y) + \frac{1}{2}p_{S_n}(x,y)$, $p(x,y)$ represents the real data distribution.

This follows from Proposition of 1 of [37] directly. Given the above result that $p_{\frac{1}{2}}(x,y) = \frac{1}{2}p_{G_n}(x,y) + \frac{1}{2}p_{S_n}(x,y)$, it is easy to verify that $p(x,y) = p_{G_n}(x,y) = p_{S_n}(x,y)$ is a global equilibrium point. However, it may not be unique and we should minimize an additional objective to ensure the uniqueness. In fact, we can achieve the uniqueness by adding a cross-entropy loss as shown in Equation (5.9).

Theorem 4. *The global equilibrium of the minimax game defined by Equation (5.9) is achieved only when $p(x,y) = p_{G_n}(x,y) = p_{S_n}(x,y)$.*

Proof. According to the definition,

$$\mathcal{L}_{ce}(S_n) = E_p[-\log p_{S_n}(y|x)] \tag{D.2}$$

which can be rewritten as:

$$D_{KL}(p(x,y)||p_{S_n}(x,y)) + H_p(y|x) \tag{D.3}$$

Namely, minimizing $\mathcal{L}_{ce}(S_n)$ is equivalent to minimizing $D_{KL}(p(x,y)||p_{S_n}(x,y))$, which is always non-negative and zero if and only if $p(x,y) = p_{S_n}(x,y)$. Besides, from Lemma 3, $p_{\frac{1}{2}}(x,y) = \frac{1}{2}p_{G_n}(x,y) + \frac{1}{2}p_{S_n}(x,y)$, $p(x,y)$, we will have $p(x,y) = p_{G_n}(x,y) = p_{S_n}(x,y)$, which concludes the proof.

D.3 More Generation and Segmentation Results on Prostate Dataset

Here, we show more experimental results that have not been shown in the main context. As we mentioned in the chapter, we conduct the same generation, segmentation, SSL-

segmentation on GalS and Prostate datasets. We show additional results on the Prostate dataset in this section.

D.3.1 Generation

First, we show the generation results on the Prostate dataset in D.1(a-c). Figure D.1(a) shows the generation process from the coarsest scale to the finest scale. Different from the GalS dataset, we used a five-level training scheme with image size of $64^2, 128^2, 256^2, 512^2, 1024^2$, which makes our final outcome to be 1024×1024. Figure D.1(b) shows three images that are generated based upon the same mask. Figure D.1(c) shows image manipulation results by changing the mask from high-grade (blue) to low-grade (green) and low-grade (green) to high-grade (blue).

D.3.2 Segmentation

Next, we implemented the fully-supervised learning on Prostate dataset. To make the results comparable with the previous literature [30, 52, 80], we used 5-fold cross validation with the standard metrics: mean Intersection Over Union (mIOU), Overall Pixel Accuracy (OPA) and Standard Mean Accuracy (SMA) to evaluate the performance of segmentation results. Assume we have segmentation results f, ground truth label l, and a pixel-wise confusion matrix $\boldsymbol{C}$, where $C_{i,j}$ is the number of pixels labeled as l_i and predicted as f_j. The mIOU is defined as the average of individual Jaccard coefficients, $\mathcal{J}_i$, for all classes l_i. The OPA is defined as the average of percent of pixels that are classified correctly for all classes l_i, $OPA = \frac{\sum_i C_{i,i}}{\sum_i \sum_j C_{i,j}}$. The standard mean accuracy is defined as $SMA = \frac{1}{N} \sum_i \frac{C_{ii}}{\sum_j C_{ij}}$. Table D.2 shows the segmentation performance of our model on the Prostate dataset. Similar to the observations in the GalS dataset, our model achieves competitive results, though not surpassing the state-of-the-art results.

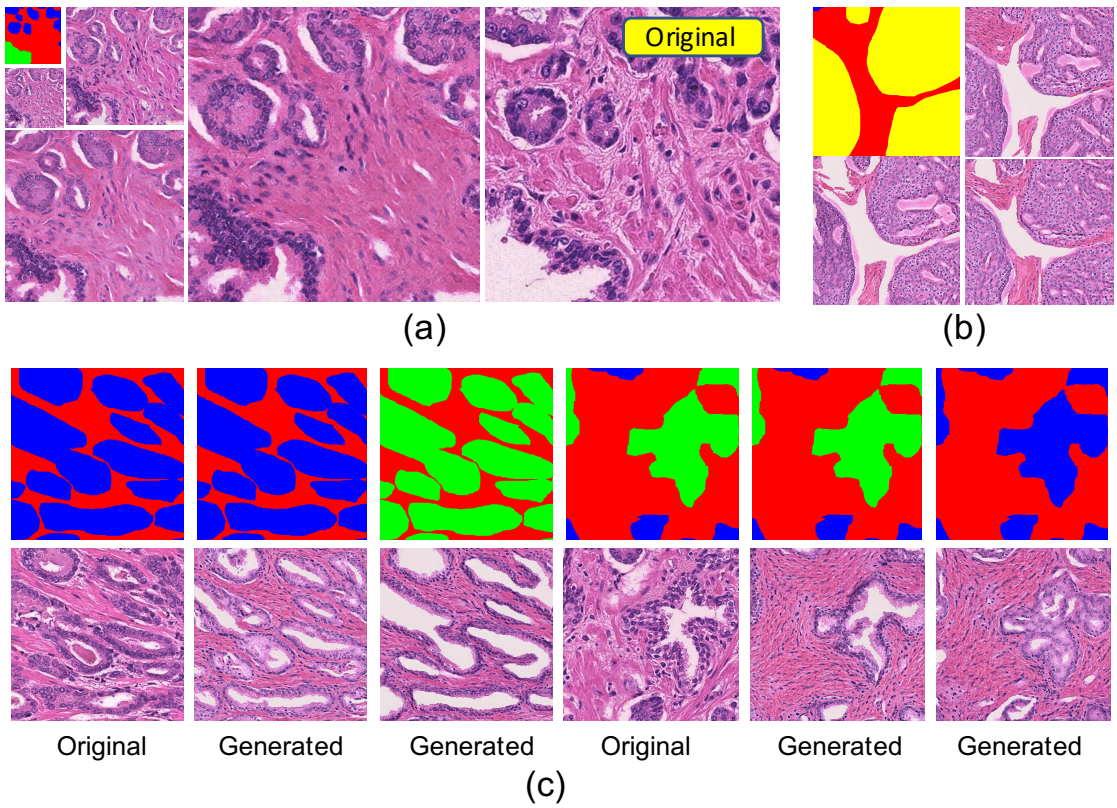

Figure D.1: (a) Generated coarse-to-fine results trained on the Prostate dataset. (b) Three generated images based on the same mask. Noise is injected during generation so that the model can synthesize images with variations. Clearer variations can been seen in the gif animation in SI. (c) Image manipulation on synthesized images. Different gland types are observed when we changed the label from low-grade (blue) to high grade (green).

D.3.3 SSL-Segmentation

We also did the similar SSL-Segmentation test on the Prostate dataset. Note that to reduce the experimental time, instead of doing 5-fold cross-validation, we kept the 20% testing dataset fixed and report our evaluation on it. For each experiment discussed in this section, we run it 5 times with different random seeds and report the mean and variance in Table D.3. As can be seen, we observe the similar trend with more substantial improvement as in GalS. Around 20% and 13% increase in mIOU are achieved compared with the *m-FCDenseNet*

Table D.2: Model performance on segmenting prostate histological images as "Stroma" (BG), "Benign" (BN), "Low-Grade" (LG), and "High-Grade" (HG). * denotes methods that are operating on 512 × 512 scale.

Method	J_{BG}	J_{BN}	J_{LG}	J_{HG}	$mIOU$	OPA	SMA
m-FCDenseNet *	0.566	0.492	0.614	0.524	0.549	0.694	0.679
Ours *	**0.826**	**0.741**	**0.713**	**0.786**	**0.767**	**0.876**	**0.872**
Handcrafted Features ([30])	0.595	0.352	0.495[1]	N/A	0.481	N/A	N/A
Multi-Scale U-Net ([80])	0.824	0.721	0.587	0.784	0.729	0.873	0.860
FCN-8s ([52])	N/A	N/A	N/A	N/A	0.759	0.873	N/A
Path R-CNN ([82])	**0.831**	**0.839**	**0.715**	**0.797**	**0.796**	**0.894**	**0.888**

Table D.3: mIOU for SSL-segmentation on Prostate dataset. All the results are generated under inductive learning unless specified in the parenthesis.

Method	20%	40%	60%	80%	100%
m-FCDenseNet	0.387 ± 0.037	0.420 ± 0.027	0.406 ± 0.025	0.492 ± 0.012	0.551 ± 0.013
m-FCDenseNet+pyramid	0.437 ± 0.051	0.500 ± 0.039	0.581 ± 0.029	0.661 ± 0.017	0.751 ± 0.009
Ours	0.527 ± 0.063	0.573 ± 0.045	0.632 ± 0.030	0.694 ± 0.021	0.767 ± 0.012
Ours (transductive)	0.577 ± 0.045	0.620 ± 0.028	0.664 ± 0.027	0.721 ± 0.013	0.787 ± 0.008

baseline for transductive and inductive learning respectively. In addition, we observe the similar trend of contributions from pyramid structure and synthetic data augmentation. G_n generated image-mask pairs are able to provide extra information gains in low-data regime, while it is negligible when we use 100% labeled data for training. Moreover, we found that the variance of model performance increases as we went to low training data regime, *i.e.* the performance variance was larger when we only used 20% training data compared to the whole training set. It indicates the importance of selected labeled data in the initial training stage.

D.4 FID Score Calculation

To calculate the Frechet Inception Distance (FID) of our synthetic images, we adopted a ResNext50 pre-trained on a large histopathology dataset. Specifically, the output layer of the model was removed and the output was taken as the activations from the last pooling layer, a global spatial pooling layer. This output layer had 1,280 activations, therefore, each image is predicted as 1,280 activation features. We applied the network on both synthetic images and real images, and got two collections of 1,280 feature vectors for them. The FID score is then calculated as follows,

$$FID = ||\mu_1 - \mu_2||^2 + Tr(C_1 + C_2 - 2(C_1 C_2)^{1/2}), \tag{D.4}$$

where $||\mu_1 - \mu_2||^2$ refers to the sum squared difference between the two mean vectors, the C_1 and C_2 are the covariance matrix for the real and synthetic feature vectors.

APPENDIX E

Appendix for Chapter 6

E.1 Pathologist Validation

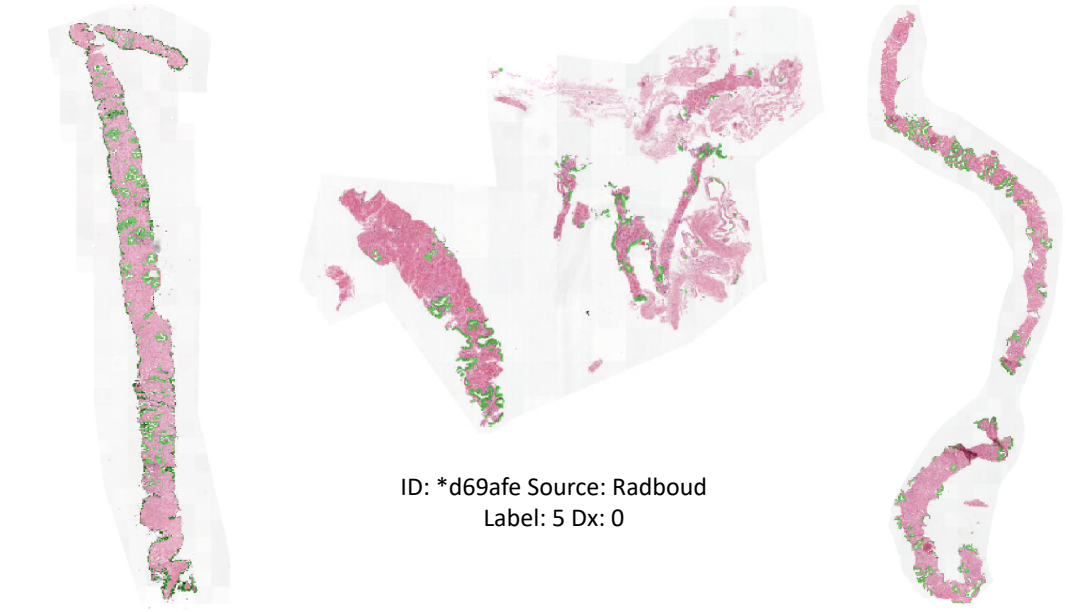

Figure E.1: Samples from "noisy" group identified by the algorithm, overlaid with the segmentation mask provided by the dataset. The mask has the following color scheme: green indicates benign, and yellow, orange and red indicate ISUP grade 3-5 respectively.

We asked a pathology expert to annotate the "easy", "noisy", and "hard" groups of samples detected by our algorithm. The pathologist would annotate these samples independently without knowing the labels provided by the dataset. As shown in the main chapter,

we found that the QWK of the "easy" group was 1, indicating 100% agreement between the pathologist and the ground-truth label for the 100 samples. Conversely, the QWK of the "noisy" group was -0.14, indicating the labels provided by the dataset were not fully consistent for these slides. The QWK of the "hard" samples was 0.10. The agreement was slightly improved compared to the "noisy group". In the following, we manually inspect example slides in each group to determine if it is reasonable that our algorithm identified them to be "noisy" and "hard" samples.

Figure E.1 shows three samples from the "noisy" group, which were annotated to be ISUP grade 0 by pathologist but labeled as ISUP grade 5 in the dataset. The slides are overlaid with the provided segmentation mask where the green color indicates benign, and yellow, orange and red indicate ISUP grade 3-5 respectively. We visually inspected the slides and did not find any suspicious cancerous areas in the slides. This was also confirmed by the segmentation mask, where no red color (ISUP grade 5) was found. Therefore, we have high confidence that these samples were annotated incorrectly, and that our algorithm identified them to be "noisy" correctly.

On the contrary, we show three samples from the "hard" group in Figure E.2. In these samples, our pathologist did not find any cancerous regions, but the ISUP grades indicated high grade cancer. By inspecting the segmentation mask, we found that only a small portion of the biopsy was overlaid with red; when zoomed in, the pathology expert could not determine the accurate ISUP grade for some slides. We argue that the disagreement between our pathologists and the dataset label is partially due to: 1) the dataset generates the ISUP grades using other information outside the biopsy, and 2) the difficult nature of Gleason grading task. Our model can successfully identify these samples to be "hard" samples that require additional inspection from the expert.

Figure E.2: Samples from "hard" group identified by the algorithm with zoomed in cancerous region.

Bibliography

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

[3] Karim Armanious, Chenming Jiang, Marc Fischer, Thomas Küstner, Tobias Hepp, Konstantin Nikolaou, Sergios Gatidis, and Bin Yang. Medgan: Medical image translation using gans. *Computerized Medical Imaging and Graphics*, 79:101684, 2020.

[4] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[5] Yaniv Bar, Idit Diamant, Lior Wolf, and Hayit Greenspan. Deep learning with non-medical training used for chest pathology identification. In *Medical Imaging 2015: Computer-Aided Diagnosis*, volume 9414, page 94140V. International Society for Optics and Photonics, 2015.

[6] Yaniv Bar, Idit Diamant, Lior Wolf, Sivan Lieberman, Eli Konen, and Hayit Greenspan. Chest pathology detection using deep learning with non-medical training. In *Biomedical Imaging (ISBI), 2015 IEEE 12th International Symposium on*, pages 294–297. IEEE, 2015.

[7] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research*, 7(Nov):2399–2434, 2006.

[8] Fadi Brimo, Rodolfo Montironi, Lars Egevad, Andreas Erbersdobler, Daniel W Lin, Joel B Nelson, Mark A Rubin, Theo Van Der Kwast, Mahul Amin, and Jonathan I Epstein. Contemporary grading for prostate cancer: implications for patient care. *European urology*, 63(5):892–901, 2013.

[9] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[10] Samuel Budd, Emma C Robinson, and Bernhard Kainz. A survey on active learning and human-in-the-loop deep learning for medical image analysis. *arXiv preprint arXiv:1910.02923*, 2019.

[11] Wouter Bulten, Hans Pinckaers, Hester van Boven, Robert Vink, Thomas de Bel, Bram van Ginneken, Jeroen van der Laak, Christina Hulsbergen-van de Kaa, and Geert Litjens. Automated gleason grading of prostate biopsies using deep learning. *arXiv preprint arXiv:1907.07980*, 2019.

[12] Hao Chen, Qi Dou, Dong Ni, Jie-Zhi Cheng, Jing Qin, Shengli Li, and Pheng-Ann Heng. Automatic fetal ultrasound standard plane detection using knowledge transferred recurrent neural networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 507–514. Springer, 2015.

[13] Hao Chen, Xiaojuan Qi, Lequan Yu, and Pheng-Ann Heng. Dcan: Deep contour-aware networks for accurate gland segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2487–2496, 2016.

[14] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[15] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018.

[16] LI Chongxuan, Taufik Xu, Jun Zhu, and Bo Zhang. Triple generative adversarial nets. In *Advances in neural information processing systems*, pages 4088–4098, 2017.

[17] Pedro Costa, Adrian Galdran, Maria Inês Meyer, Michael David Abràmoff, Meindert Niemeijer, Ana Maria Mendonça, and Aurélio Campilho. Towards adversarial retinal image synthesis. *arXiv preprint arXiv:1701.08974*, 2017.

[18] Gabriela Csurka, Diane Larlus, Florent Perronnin, and France Meylan. What is a good evaluation measure for semantic segmentation?. In *BMVC*, volume 27, page 2013. Citeseer, 2013.

[19] Zihang Dai, Zhilin Yang, Fan Yang, William W Cohen, and Ruslan R Salakhutdinov. Good semi-supervised learning that requires a bad gan. In *Advances in neural information processing systems*, pages 6510–6520, 2017.

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

[21] Yair Dgani, Hayit Greenspan, and Jacob Goldberger. Training a neural network based on unreliable human annotation of medical images. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 39–42. IEEE, 2018.

[22] Scott Doyle, Michael Feldman, John Tomaszewski, and Anant Madabhushi. A boosted bayesian multiresolution classifier for prostate cancer detection from digitized needle biopsies. *IEEE transactions on biomedical engineering*, 59(5):1205–1218, 2012.

[23] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

[24] Jonathan I Epstein. Prostate biopsy interpretation. *Biopsy Interpretation Series*, 1995.

[25] Jonathan I Epstein, William C Allsbrook Jr, Mahul B Amin, Lars L Egevad, ISUP Grading Committee, et al. The 2005 international society of urological pathology (isup) consensus conference on gleason grading of prostatic carcinoma. *The American journal of surgical pathology*, 29(9):1228–1242, 2005.

[26] Reza Farjam, Hamid Soltanian-Zadeh, Reza Aghaizadeh Zoroofi, and Kourosh Jafari-Khouzani. Tree-structured grading of pathological images of prostate. In *Medical Imaging 2005: Image Processing*, volume 5747, pages 840–852. International Society for Optics and Photonics, 2005.

[27] Samson W Fine, Mahul B Amin, Daniel M Berney, Anders Bjartell, Lars Egevad, Jonathan I Epstein, Peter A Humphrey, Christina Magi-Galluzzi, Rodolfo Montironi, and Christian Stief. A contemporary update on pathology reporting for prostate cancer: biopsy and radical prostatectomy specimens. *European urology*, 62(1):20–39, 2012.

[28] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. *arXiv preprint arXiv:1703.02910*, 2017.

[29] Zhe Gan, Liqun Chen, Weiyao Wang, Yuchen Pu, Yizhe Zhang, Hao Liu, Chunyuan Li, and Lawrence Carin. Triangle generative adversarial networks. In *Advances in Neural Information Processing Systems*, pages 5247–5256, 2017.

[30] Arkadiusz Gertych, Nathan Ing, Zhaoxuan Ma, Thomas J Fuchs, Sadri Salman, Sambit Mohanty, Sanica Bhele, Adriana Velásquez-Vacca, Mahul B Amin, and Beatrice S Knudsen. Machine learning approaches to analyze histological images of tissues from radical prostatectomies. *Computerized Medical Imaging and Graphics*, 46:197–208, 2015.

[31] Sanjay Surendranath Girija. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *Software available from tensorflow. org*, 2016.

[32] Ross Girshick. Fast r-cnn. *arXiv preprint arXiv:1504.08083*, 2015.

[33] 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.

[34] Donald F Gleason. Histologic grading of prostate cancer: a perspective. *Human pathology*, 23(3):273–279, 1992.

[35] Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. 2016.

[36] Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

[37] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[38] Lena Gorelick, Olga Veksler, Mena Gaed, José A Gómez, Madeleine Moussa, Glenn Bauman, Aaron Fenster, and Aaron D Ward. Prostate histopathology: Learning tissue component histograms for cancer detection and classification. *IEEE transactions on medical imaging*, 32(10):1804–1818, 2013.

[39] Simon Graham, David Epstein, and Nasir Rajpoot. Rota-net: Rotation equivariant network for simultaneous gland and lumen segmentation in colon histology images. In *European Congress on Digital Pathology*, pages 109–116. Springer, 2019.

[40] Hayit Greenspan, Bram van Ginneken, and Ronald M Summers. Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique. *IEEE Transactions on Medical Imaging*, 35(5):1153–1159, 2016.

[41] John T Guibas, Tejpal S Virdi, and Peter S Li. Synthetic medical images from dual generative adversarial networks. *arXiv preprint arXiv:1709.01872*, 2017.

[42] Sheng Guo, Weilin Huang, Haozhi Zhang, Chenfan Zhuang, Dengke Dong, Matthew R Scott, and Dinglong Huang. Curriculumnet: Weakly supervised learning from large-scale web images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 135–150, 2018.

[43] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in neural information processing systems*, pages 8527–8537, 2018.

[44] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017.

[45] 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.

[46] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pages 6626–6637, 2017.

[47] Mitch Hill, Erik Nijkamp, and Song-Chun Zhu. Building a telescope to look into high-dimensional image spaces. *arXiv preprint arXiv:1803.01043*, 2018.

[48] N. Houlsby, Ferenc Huszár, Zoubin Ghahramani, and M. Lengyel. Bayesian active learning for classification and preference learning. *ArXiv*, abs/1112.5745, 2011.

[49] Cheng Cheng Huang, Max Xiangtian Kong, Ming Zhou, Andrew B Rosenkrantz, Samir S Taneja, Jonathan Melamed, and Fang-Ming Deng. Gleason score 3+ 4= 7 prostate cancer with minimal quantity of gleason pattern 4 on needle biopsy is associated with low-risk tumor in radical prostatectomy specimen. *The American journal of surgical pathology*, 38(8):1096–1101, 2014.

[50] Jinchi Huang, Lie Qu, Rongfei Jia, and Binqiang Zhao. O2u-net: A simple noisy label detection approach for deep neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3326–3334, 2019.

[51] Peter A Humphrey. Gleason grading and prognostic factors in carcinoma of the prostate. *Modern pathology*, 17(3):292, 2004.

[52] Nathan Ing, Zhaoxuan Ma, Jiayun Li, Hootan Salemi, Corey Arnold, Beatrice S Knudsen, and Arkadiusz Gertych. Semantic segmentation for prostate cancer grading by convolutional neural networks. In *Medical Imaging 2018: Digital Pathology*, volume 10581, page 105811B. International Society for Optics and Photonics, 2018.

[53] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5070–5079, 2019.

[54] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[55] Kourosh Jafari-Khouzani and Hamid Soltanian-Zadeh. Multiwavelet grading of pathological images of prostate. *IEEE Transactions on Biomedical Engineering*, 50(6):697–704, 2003.

[56] Andrew Janowczyk, Ajay Basavanhally, and Anant Madabhushi. Stain normalization using sparse autoencoders (stanosa): application to digital pathology. *Computerized Medical Imaging and Graphics*, 57:50–61, 2017.

[57] Andrew Janowczyk and Anant Madabhushi. Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases. *Journal of pathology informatics*, 7, 2016.

[58] Simon Jégou, Michal Drozdzal, David Vazquez, Adriana Romero, and Yoshua Bengio. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 11–19, 2017.

[59] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*, pages 2304–2313, 2018.

[60] Konstantinos Kamnitsas, Christian Baumgartner, Christian Ledig, Virginia Newcombe, Joanna Simpson, Andrew Kane, David Menon, Aditya Nori, Antonio Criminisi, Daniel Rueckert, et al. Unsupervised domain adaptation in brain lesion segmentation with adversarial networks. In *International conference on information processing in medical imaging*, pages 597–609. Springer, 2017.

[61] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

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

[63] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[64] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. *arXiv preprint arXiv:1703.04730*, 2017.

[65] Daisuke Komura and Shumpei Ishikawa. Machine learning methods for histopathological image analysis. *Computational and structural biotechnology journal*, 16:34–42, 2018.

[66] Ksenia Konyushkova, Raphael Sznitman, and Pascal Fua. Geometry in active learning for binary and multi-class image segmentation. *Computer vision and image understanding*, 182:1–16, 2019.

[67] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pages 109–117, 2011.

[68] Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Andreas Veit, et al. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://github. com/openimages*, 2:3, 2017.

[69] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[70] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[71] Abhishek Kumar, Prasanna Sattigeri, and Tom Fletcher. Semi-supervised learning with gans: Manifold invariance with improved inference. In *Advances in Neural Information Processing Systems*, pages 5534–5544, 2017.

[72] Neeraj Kumar, Ruchika Verma, Sanuj Sharma, Surabhi Bhargava, Abhishek Vahadane, and Amit Sethi. A dataset and a technique for generalized nuclear segmentation for computational pathology. *IEEE transactions on medical imaging*, 36(7):1550–1560, 2017.

[73] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

[74] Hugh J Lavery and Michael J Droller. Do gleason patterns 3 and 4 prostate cancer represent separate disease states? *The Journal of urology*, 188(5):1667–1675, 2012.

[75] Bruno Lecouat, Chuan-Sheng Foo, Houssam Zenati, and Vijay R Chandrasekhar. Semi-supervised learning with gans: Revisiting manifold regularization. *arXiv preprint arXiv:1805.08957*, 2018.

[76] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[77] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[78] Kuang-Huei Lee, Xiaodong He, Lei Zhang, and Linjun Yang. Cleannet: Transfer learning for scalable image classifier training with label noise. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5447–5456, 2018.

[79] Jiayun Li, Karthik V. Sarma, King Chung Ho, Arkadiusz Gertych, Beatrice S. Knudsen, and Corey W. Arnold. A Multi-scale U-Net for Semantic Segmentation of Histological Images from Radical Prostatectomies. In *AMIA Annual Symposium Proceedings*, 2017.

[80] Jiayun Li, Karthik V Sarma, King Chung Ho, Arkadiusz Gertych, Beatrice S Knudsen, and Corey W Arnold. A multi-scale u-net for semantic segmentation of histological images from radical prostatectomies. In *AMIA Annual Symposium Proceedings*, volume 2017, page 1140. American Medical Informatics Association, 2017.

[81] Jiayun Li, William Speier, Karthik V Sarma, King Chung Ho, Arkadiusz Gertych, Beatrice S Knudsen, and Corey W Arnold. An em-based semi-supervised deep learning approach for semantic segmentation of histopathological images from radical prostatectomies. *Computerized Medical Imaging and Graphics*, 2018.

[82] Wenyuan Li, Jiayun Li, Karthik V Sarma, King Chung Ho, Shiwen Shen, Beatrice S Knudsen, Arkadiusz Gertych, and Corey W Arnold. Path r-cnn for prostate cancer diagnosis and gleason grading of histological images. *IEEE transactions on medical imaging*, 38(4):945–954, 2018.

[83] Wenyuan Li, Yunlong Wang, Yong Cai, Corey Arnold, Emily Zhao, and Yilian Yuan. Semi-supervised rare disease detection using generative adversarial network. *arXiv preprint arXiv:1812.00547*, 2018.

[84] Wenyuan Li, Zichen Wang, Jiayun Li, Jennifer Polson, William Speier, and Corey Arnold. Semi-supervised learning based on generative adversarial network: a comparison between good gan and bad gan approach. *arXiv preprint arXiv:1905.06484*, 2019.

[85] Yuexiang Li and Linlin Shen. cc-gan: A robust transfer-learning framework for hep-2 specimen image segmentation. *IEEE Access*, 6:14048–14058, 2018.

[86] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, volume 1, page 4, 2017.

[87] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[88] 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.

[89] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[90] Mario Lucic, Michael Tschannen, Marvin Ritter, Xiaohua Zhai, Olivier Bachem, and Sylvain Gelly. High-fidelity image generation with fewer labels. *arXiv preprint arXiv:1903.02271*, 2019.

[91] L. V. D. Maaten and Geoffrey E. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[92] Ali Madani, Mehdi Moradi, Alexandros Karargyris, and Tanveer Syeda-Mahmood. Semi-supervised learning with generative adversarial networks for chest x-ray classification with ability of data domain adaptation. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 1038–1042. IEEE, 2018.

[93] Faisal Mahmood, Daniel Borders, Richard Chen, Gregory N McKay, Kevan J Salimian, Alexander Baras, and Nicholas J Durr. Deep adversarial training for multi-organ nuclei segmentation in histopathology images. *IEEE transactions on medical imaging*, 2019.

[94] Inc. Matterport. Mask r-cnn for object detection and instance segmentation on keras and tensorflow. https://github.com/matterport/Mask_RCNN, 2018.

[95] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[96] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[97] Takeru Miyato, Shin-ichi Maeda, Shin Ishii, and Masanori Koyama. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[98] Tony CW Mok and Albert CS Chung. Learning data augmentation for brain tumor segmentation with coarse-to-fine generative adversarial networks. In *International MICCAI Brainlesion Workshop*, pages 70–80. Springer, 2018.

[99] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *Advances in neural information processing systems*, 2011.

[100] Kien Nguyen, Bikash Sabata, and Anil K Jain. Prostate cancer grading: Gland segmentation and structural features. *Pattern Recognition Letters*, 33(7):951–961, 2012.

[101] Kamal Nigam, Andrew McCallum, and Tom Mitchell. Semi-supervised text classification using em. *Semi-Supervised Learning*, pages 33–56, 2006.

[102] Firat Ozdemir, Zixuan Peng, Christine Tanner, Philipp Fuernstahl, and Orcun Goksel. Active learning for segmentation by optimizing content information for maximal entropy. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 183–191. Springer, 2018.

[103] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *Advances in neural information processing systems Workshop*, 2017.

[104] Yunchen Pu, Zhe Gan, Ricardo Henao, Xin Yuan, Chunyuan Li, Andrew Stevens, and Lawrence Carin. Variational autoencoder for deep learning of images, labels and captions. In *Advances in neural information processing systems*, pages 2352–2360, 2016.

[105] Siyuan Qiao, Wei Shen, Zhishuai Zhang, Bo Wang, and Alan Yuille. Deep co-training for semi-supervised image recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 135–152, 2018.

[106] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[107] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. In *Advances in neural information processing systems*, pages 3546–3554, 2015.

[108] Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014.

[109] Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE Computer graphics and applications*, 21(5):34–41, 2001.

[110] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[111] 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.

[112] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Mutual exclusivity loss for semi-supervised deep learning. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 1908–1912. IEEE, 2016.

[113] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

[114] Caglar Senaras, Muhammad Khalid Khan Niazi, Berkman Sahiner, Michael P Pennell, Gary Tozbikian, Gerard Lozanski, and Metin N Gurcan. Optimized generation of high-resolution phantom images using cgan: Application to quantification of ki67 breast cancer images. *PloS one*, 13(5), 2018.

[115] Amit Sethi, Lingdao Sha, Abhishek Ramnath Vahadane, Ryan J Deaton, Neeraj Kumar, Virgilia Macias, and Peter H Gann. Empirical comparison of color normalization methods for epithelial-stromal classification in h and e images. *Journal of pathology informatics*, 7, 2016.

[116] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. Singan: Learning a generative model from a single natural image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4570–4580, 2019.

[117] Hoo-Chang Shin, Holger R Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura, and Ronald M Summers. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, 35(5):1285–1298, 2016.

[118] Rebecca L Siegel, Kimberly D Miller, and Ahmedin Jemal. Cancer statistics, 2016. *CA: a cancer journal for clinicians*, 66(1):7–30, 2016.

[119] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[120] Korsuk Sirinukunwattana, Josien PW Pluim, Hao Chen, Xiaojuan Qi, Pheng-Ann Heng, Yun Bo Guo, Li Yang Wang, Bogdan J Matuszewski, Elia Bruni, Urko Sanchez, et al. Gland segmentation in colon histology images: The glas challenge contest. *Medical image analysis*, 35:489–502, 2017.

[121] Korsuk Sirinukunwattana, Shan E Ahmed Raza, Yee-Wah Tsang, David RJ Snead, Ian A Cree, and Nasir M Rajpoot. Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images. *IEEE transactions on medical imaging*, 35(5):1196–1206, 2016.

[122] Asim Smailagic, Hae Young Noh, Pedro Costa, Devesh Walawalkar, Kartik Khandelwal, Mostafa Mirshekari, Jonathon Fagert, Adrián Galdrán, and Susu Xu. Medal: Deep active learning sampling method for medical image analysis. *arXiv preprint arXiv:1809.09287*, 2018.

[123] Yoav Smith, Gershom Zajicek, Michael Werman, Galina Pizov, and Yoav Sherman. Similarity measurement method for the classification of architecturally differentiated images. *Computers and Biomedical Research*, 32(1):1–12, 1999.

[124] Jamshid Sourati, Ali Gholipour, Jennifer G Dy, Sila Kurugol, and Simon K Warfield. Active deep learning with fisher information for patch-wise semantic segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 83–91. Springer, 2018.

[125] Rachel Sparks and Anant Madabhushi. Out-of-sample extrapolation utilizing semi-supervised manifold learning (ose-ssl): content based image retrieval for histopathology images. *Scientific reports*, 6:27306, 2016.

[126] William Speier, Jiayun Li, Wenyuan Li, Karthik Sarma, and Corey Arnold. Image-based patch selection for deep learning to improve automated gleason grading in histopathological slides. *bioRxiv*, 2020.

[127] Jost Tobias Springenberg. Unsupervised and semi-supervised learning with categorical generative adversarial networks. *arXiv preprint arXiv:1511.06390*, 2015.

[128] Rainer Stotzka, Reinhard Männer, Peter H Bartels, and Deborah Thompson. A hybrid neural and statistical classifier system for histopathologic grading of prostatic lesions. *Analytical and quantitative cytology and histology*, 17(3):204–218, 1995.

[129] Peter Szolovits, Ramesh S Patil, and William B Schwartz. Artificial intelligence in medical diagnosis. *Annals of internal medicine*, 108(1):80–87, 1988.

[130] Ali Tabesh, Mikhail Teverovskiy, Ho-Yuen Pang, Vinay P Kumar, David Verbel, Angeliki Kotsianti, and Olivier Saidi. Multifeature prostate cancer diagnosis and gleason grading of histological images. *IEEE transactions on medical imaging*, 26(10):1366–1378, 2007.

[131] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019.

[132] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204, 2017.

[133] Dayong Wang, Aditya Khosla, Rishab Gargeya, Humayun Irshad, and Andrew H Beck. Deep learning for identifying metastatic breast cancer. *arXiv preprint arXiv:1606.05718*, 2016.

[134] 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.

[135] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. pix2pixhd: High-resolution image synthesis and semantic manipulation with conditional gans.

[136] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018.

[137] Si Wen, Tahsin M Kurc, Le Hou, Joel H Saltz, Rajarsi R Gupta, Rebecca Batiste, Tianhao Zhao, Vu Nguyen, Dimitris Samaras, and Wei Zhu. Comparison of different classifiers with active learning to support quality control in nucleus segmentation in pathology images. *AMIA Summits on Translational Science Proceedings*, 2018:227, 2018.

[138] Jason Weston, Frédéric Ratle, Hossein Mobahi, and Ronan Collobert. Deep learning via semi-supervised embedding. In *Neural Networks: Tricks of the Trade*, pages 639–655. Springer, 2012.

[139] Arthur W Wetzel, R Crowley, Sujin Kim, R Dawson, Lei Zheng, YM Joo, Yukako Yagi, John Gilbertson, C Gadd, DW Deerfield, et al. Evaluation of prostate tumor grades by content-based image retrieval. In *27th AIPR Workshop: Advances in Computer-Assisted Recognition*, volume 3584, pages 244–253. International Society for Optics and Photonics, 1999.

[140] Wikipedia contributors. Gleason grading system — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Gleason_grading_system&oldid=827248377, 2018. [Online; accessed 14-August-2018].

[141] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2691–2699, 2015.

[142] Cheng Xue, Qi Dou, Xueying Shi, Hao Chen, and Pheng-Ann Heng. Robust learning at noisy labeled medical images: Applied to skin lesion classification. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pages 1280–1283. IEEE, 2019.

[143] Yuan Xue, Tao Xu, Han Zhang, L Rodney Long, and Xiaolei Huang. Segan: Adversarial network with multi-scale l 1 loss for medical image segmentation. *Neuroinformatics*, 16(3-4):383–392, 2018.

[144] Ching-Wei Yang, Tzu-Ping Lin, Yi-Hsiu Huang, Hsiao-Jen Chung, Junne-Yih Kuo, William JS Huang, Howard HH Wu, Yen-Hwa Chang, Alex TL Lin, and Kuang-Kuo Chen. Does extended prostate needle biopsy improve the concordance of gleason scores between biopsy and prostatectomy in the taiwanese population? *Journal of the Chinese Medical Association*, 75(3):97–101, 2012.

[145] Junlin Yang, Nicha C Dvornek, Fan Zhang, Julius Chapiro, MingDe Lin, and James S Duncan. Unsupervised domain adaptation via disentangled representations: Application to cross-modality liver segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 255–263. Springer, 2019.

[146] Lin Yang, Yizhe Zhang, Jianxu Chen, Siyuan Zhang, and Danny Z Chen. Suggestive annotation: A deep active learning framework for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 399–407. Springer, 2017.

[147] Xin Yi, Ekta Walia, and Paul Babyn. Generative adversarial network in medical imaging: A review. *Medical image analysis*, page 101552, 2019.

[148] Chenyu You, Guang Li, Yi Zhang, Xiaoliu Zhang, Hongming Shan, Mengzhou Li, Shenghong Ju, Zhen Zhao, Zhuiyang Zhang, Wenxiang Cong, et al. Ct super-resolution gan constrained by the identical, residual, and cycle learning ensemble (gan-circle). *IEEE Transactions on Medical Imaging*, 39(1):188–203, 2019.

[149] Yizhe Zhang, Lin Yang, Jianxu Chen, Maridel Fredericksen, David P Hughes, and Danny Z Chen. Deep adversarial networks for biomedical image segmentation utilizing unannotated images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 408–416. Springer, 2017.