LARGE-SCALE DEEP LEARNING WITH APPLICATION IN MEDICAL IMAGING AND BIO-INFORMATICS

by

ZHENG XU

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON
December 2018

To my family, for their endless trust, support, encourage and love.

ACKNOWLEDGEMENTS

There were many people who helped me during my PhD studying career, and I would like to take this opportunity to thank them.

I would like to thank my supervising professor Dr. Junzhou Huang for constantly motivating and encouraging me, and also for his invaluable advice during the course of my doctoral studies. He held me to the highest of standards, but also had the faith that I would be able to achieve them. None of the work in this thesis would have happened without him.

I wish to thank my thesis committee members Dr. Hong Jiang, Dr. Chris Ding, Dr. Dajiang Zhu for their interest in my research and for their valuable suggestions regarding my early proposal and this thesis. It is a privilege for me to have each of them serve in my committees

The research described in this thesis has benefited from my other collaborators besides my advisors. Without them, some of the chapters in this thesis would not have been possible. My special thanks go to Dr. Heng Huang, Prof. Leon Axel and Prof. Xiaolei Huang. I have been learning a lot from them through the collaborations.

I want to thank all my colleagues from the Scalable Modeling and Imaging and Learning Lab (SMILE), the Computer Science and Engineering Department. It is my pleasure to meet such a concentration of creative and nice people here. I am grateful to all with whom I spent my time as a graduate student at UTA.

Finally, my special thanks go to my family. I would like to express my earnest gratitude to my parents for their love and countless sacrifices to give me the best

possible education. Without their patience and unreserved support, it would not have been possible to reach this stage in my career.

November 1, 2018

ABSTRACT

LARGE-SCALE DEEP LEARNING WITH APPLICATION IN MEDICAL IMAGING AND BIO-INFORMATICS

ZHENG XU, Ph.D.

The University of Texas at Arlington, 2018

Supervising Professor: Junzhou Huang

With the recent advancement of the **deep learning** technology in the artificial intelligence area, nowadays people's lives have been drastically changed. However, the success of deep learning technology mostly relies on large-scale high-quality data-sets. The complexity of deeper model and larger scale datasets have brought us significant challenges. Inspired by this trend, in this dissertation, we focus on developing efficient and effective large-scale deep learning techniques in solving real-world problems, like cell detection in hyper-resolution medical image or drug screening from millions of compound candidates.

With respect to the hyper-resolution medical imaging cell detection problem, the challenges are mainly the extremely large scale pixel information. Also the cell density in the region of interests are usually super high, meaning that the cells will clump and congest in small areas. These challenges hence demand high quality efficient modeling to address this cell detection problem at scale. In this paper, we will discuss the large-scale cell detection problem from both mathematical/statistical

modeling and architectural system perspective and reach to a comprehensive solution, which is both incredibly efficient and effective.

With respect to the drug discovery problem, every drug company with R&D department has carried out numerous initiatives for speeding up its drug discovery process. Drug discovery is the process through which potential new medicines are identified. Modern drug discovery is usually implemented as drug compound selection, while, for every candidate chemical compound, the chemical drug properties, e.g., affinity, selectivity, metabolic stability, are biologically tested in the lab environment. Once all the properties pass the drug requirement tests, it will be selected as a new potential drug candidate. However, this process is excessively expensive and labor-intensive, and costs hundreds of million dollars each year. The major challenge for deep learning is to take in the sequence representation of drug compound, i.e, SMILE representation as input and infer chemical properties from limited high-quality dataset. Within this context, we propose several effective unsupervised/semi-supervised techniques in generating the powerful chemical representation and models that provide strong inference.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

This thesis focus on developing large-scale deep learning techniques for the purpose of handling medical imaging and bio-informatics tasks, e.g. cell detection, drug discovery, etc.

1.1 Motivation

First of all, we focus on the automatic lung cancer cell detection problem using deep learning techniques [2, 3]. Automatic lung cancer cell detection is the basis of many computer-assisted methods for cell-based experiments and diagnosis. However, at present, very few work has been focused on lung cancer cell detection. The difficulty in lung cancer cell detection problem is basically three-fold. First, the density of lung tumor cells is generally very high in the histopathological images. Second, the cell size might vary and cell clumping is usual. Third, the time cost of cell detection method, especially in high-resolution histopathological images, is very high in cell-based diagnosis. With these challenges mentioned above, it is still in great demand for researchers to develop efficient and robust lung cancer cell detection methods. To alleviate these problems, we propose an efficient and robust lung cancer cell detection method based on the Deep Convolution Neural Network(DCNN)[4]. Other than computationally-intensive frameworks [5, 6], or ROI(region of interest)-based detection method[1, 7], it exploits the deep architecture to learn the hierarchical discriminative features, which has recently achieved significant success in biomedical image analysis[8, 9].

Secondly, we investigate the problem of handling small-scale labeled drug discovery data with large-scale unlabeled drug discovery data [10, 11]. Specifically, iN the past few years, the application of Artificial Intelligence (AI) technologies in drug discovery has become significant and increasingly popular. Observing the most recent rapid growth of a key technology in AI, namely **deep learning** (or **deep neural network**), the whole industry and academia are looking towards AI to speed up the drug discovery, cut R&D cost and decrease the failure rate in potential drug screening trials [12].

However, the previous success of deep learning in multiple applications, e.g., image understanding [13, 14], medical imaging [15, 3, 16], video understanding [17, 18], bioinformatics [10, 19, 20], and machine translation [21], etc., has implied a reliance on large-scale high-quality labeled data-sets. The training procedure of those deep-learning-based state-of-the-art models generally involve millions of labeled samples. In the meantime, however, for the drug discovery tasks, the scale of labeled data-set stays around only thousands of examples due to the insanely high cost of obtaining the clean labeled data through the biological experiments. The available amount of the labeled training data is absolutely insufficient to secure the success of the application of deep learning in the drug discovery. This huge gap between the requirement and availability of the labeled data in drug discovery has become a bottleneck of applying deep learning techniques into drug discovery.

1.2 Our Techniques

In the original problem of large-scale cell segmentation, we propose a novel DCNN based model for lung cancer cell detection in this paper. Our contributions are summarized as three parts: 1) We built up a deep learning-based framework in lung cancer cell detection with modified sliding window manner in both training and testing stage. 2) We modify the training strategy by only acquiring weak annotations in the samples, which decreases both labeling and training cost. 3) We present a novel accelerated DCNN forwarding technology by reducing the redundant convolution computation, accelerating the testing process several hundred times than the traditional DCNN-based sliding window method. To the best of our knowledge, this is the first study to report the application of accelerated DCNN framework for lung cancer cell detection.

To the best of our knowledge, the research presented in this paper represents the first attempt to develop an extremely efficient deep neural network based pixel-wise cell detection framework for whole-slide images. Particularly, it is general enough to cooperate with any deep convolutional neural networks to work on whole-slide imaging. Our technical contributions are summarized as: 1) A general sparse kernel neural network model is applied for the pixel-wise cell detection, accelerating the forwarding procedure of the deep convolutional neural networks. 2) An asynchronous prefetching technique is proposed to reduce nearly 95% of the disk I/O time. 3) We propose a scalable and communication efficient framework to extend our neural network to multi-GPU and cluster environments, dramatically accelerating the entire detecting process. Extensive experiments have been conducted to demonstrate the efficiency and effectiveness of our method.

In this paper, for drug discovery problem, we propose an unsupervised data-driven deep-learning-based molecular fingerprint method, named **seq2seq fingerprint**. To overcome the issues mentioned above, 1) the proposed method is data-driven, without any human expertise knowledge required. 2) the fingerprints generated by the proposed method are completely revertible to original molecular representations, ensuring the sufficiency of information encoded in the fingerprint vector. 3) the proposed method employs an unsupervised training on a **huge unlabeled** dataset, sufficiently releasing the horsepower of deep neural network. We illustrate a comparison among all mentioned fingerprint methods and our seq2seq fingerprint method.

Furthermore, to wield the power of the supervised learning, we proposed seq3seq fingerprint framework. The seq3seq fingerprint network can be considered as a pipeline with one input and two outputs. The designed neural network can take the molecule inputs for training, **with or without labels**. The input is the raw sequence representation of a molecule, namely SMILE representation. The two outputs will correspond to the two tasks inside this network. The first one is the **self-recovery**. The network is expected to be able to generate a vector representation which is able to be recovered back to original raw sequence representation. The second task is the **inference** whenever the label is available. For instance, it can be a task to predict the acidity, alkalinity or solubility of a single molecule. The two tasks are trained within the same network in an end-to-end fashion. As a result, in a specific inference task, the vector representation will be able to provide both good recovery performance and inference performance. Also, the network can be trained inside a mixture data pool with both labeled and unlabeled data, which is sufficient enough to ensure the fine training of the neural network.

1.3 Thesis Overview

Finally, we provide the overview of this thesis in brief. In Chapter 2, we present our efficient deep learning modeling approach to handle large-scale whole-slide image for cell detection/segmentation task. Then, Chapter 3 generalize the deep learning approach to a broader context to make it more practical in distributed computing environment. Chapter 4 presents our unsupervised learning-based embedding approach (seq2seq fingerprint) on another kind of data: sequence-based drug discovery data.

Then, Chapter 5 presents the semi-supervised version of the seq2seq fingerprint, which brings a relatively small amount of supervised data into the modeling and improve the inference performance significantly.

As the ending, Chapter 6 draws our conclusions of the thesis, where we summarize the presented large-scale deep learning techniques, highlight their contributions in both theory and practice, and provide some future research directions.

CHAPTER 2

EFFICIENT LUNG CANCER CELL DETECTION WITH DEEP CONVOLUTION NEURAL NETWORK

Lung cancer cell detection serves as an important step in the automation of cell-based lung cancer diagnosis. In this paper, we propose a robust and efficient lung cancer cell detection method based on the accelerated Deep Convolution Neural Network framework(DCNN). The efficiency of the proposed method is demonstrated in two aspects: 1) We adopt a training strategy, learning the DCNN model parameters from only weakly annotated cell information (one click near the nuclei location). This technique significantly reduces the manual annotation cost and the training time. 2) We introduce a novel DCNN forward acceleration technique into our method, which speeds up the cell detection process several hundred times than the conventional sliding-window based DCNN. In the reported experiments, state-of-the-art accuracy and the impressive efficiency are demonstrated in the lung cancer histopathological image dataset. [2]

2.1 Introduction

Automatic lung cancer cell detection is the basis of many computer-assisted methods for cell-based experiments and diagnosis. However, at present, very few work has been focused on lung cancer cell detection. The difficulty in lung cancer cell detection problem is basically three-fold. First, the density of lung tumor cells is generally very high in the histopathological images. Second, the cell size might vary and cell clumping is usual. Third, the time cost of cell detection method, especially in high-resolution histopathological images, is very high in cell-based diagnosis. With these challenges mentioned above, it is still in great demand for researchers to develop efficient and robust lung cancer cell detection methods. To alleviate these problems, we propose an efficient and robust lung cancer cell detection method based on the Deep Convolution Neural Network(DCNN)[4]. Other than computationally-intensive frameworks [5, 6], or ROI(region of interest)-based detection method[1, 7], it exploits the deep architecture to learn the hierarchical discriminative features, which has recently achieved significant success in biomedical image analysis[8, 9].

In the proposed method, the training process is only performed on the local patches centered at the weakly annotated dot in each cell area with the non-cell area patches of the same amount as the cell areas. This means only weak annotation of cell area (a single dot near the center of cell area) are required during labeling process, significantly relieving the manual annotation burden. Another benefit for this technique is to reduce the over-fitting effect and make the proposed method general enough to detect the rough cell shape information in the training image, providing the benefit for further applications, e.g. cell counting, segmentation and tracking.

During testing stage, the conventional sliding window manner for all local pixel patches is inefficient due to the considerable redundant convolution computation. To accelerate the testing process for each testing image, we present a fast forwarding technique in DCNN framework. Instead of preforming DCNN forwarding in each pixel patch, the proposed method performs convolution computation in the entire testing image, with a modified sparse convolution kernel. This technique almost eliminates all redundant convolution computation compared to the conventional pixel-wise classification, which significantly accelerates the DCNN forwarding procedure. Experimental result reports the proposed method only requires around 0.1 second to detect lung cancer cells in a 512 × 512 image, while the state-of-the-art DCNN requires around 40 seconds.

To sum up, we propose a novel DCNN based model for lung cancer cell detection in this paper. Our contributions are summarized as three parts: 1) We built up a deep learning-based framework in lung cancer cell detection with modified sliding window manner in both training and testing stage. 2) We modify the training strategy by only acquiring weak annotations in the samples, which decreases both labeling and training cost. 3) We present a novel accelerated DCNN forwarding technology by reducing the redundant convolution computation, accelerating the testing process several hundred times than the traditional DCNN-based sliding window method. To the best of our knowledge, this is the first study to report the application of accelerated DCNN framework for lung cancer cell detection.

2.2 Methodology

Given an input lung cancer histopathological image I, the problem is to find a set $D = \{d_1, d_2, \ldots, d_N\}$ of detections, each reporting the centroid coordinates for a single cell area. The problem is solved by training a detector on training images with given weakly annotated ground truth information $G = \{g_1, g_2, \ldots, g_M\}$, each representing the manually annotated coordinate near the center of each cell area. In the testing stage, each pixel is assigned one of two possible classes, *cell* or *non-cell*, the former to pixels in cell areas, the latter to all other pixels. Our detector is a DCNN-based pixel-wise classifier. For each given pixel p, the DCNN predicts its class using raw RGB values in its local square image patch centered on p.

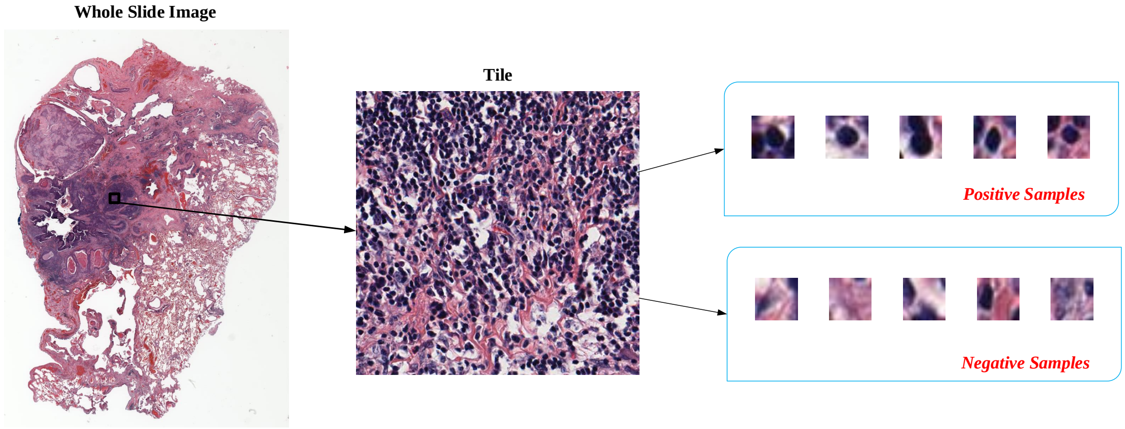


Figure 2.1: The illustration of generation of training samples: 1) Tiles are randomly sampled from the whole slide images. 2) The sampled tiles are manually annotated by well-trained pathologists, which construct the weakly annotated information. 3) We only feed the local pixels patches center on the annotated pixels and the randomly sampled non-cell patches of the same amount as the cell ones.

2.2.1 Training the detector

Using the weakly annotated ground truth data G, we label each patch centered on the given ground truth g_m as positive($cell$) sample. Moreover, we randomly sample the negative(non-$cell$) samples from the local pixel patches whose center are outside of the boundary of positive patches. The amount of negative sample patches is the same as the positive ones. If a patch window lies partly outside of the image boundary, the missing pixels are fetched in the mirror padded image.

For these images, we only feed very few patches into the proposed model for training, therefore extremely accelerating the training stage. Besides, this technique also partly eliminates the effect of over-fitting due to the under-sampling usage of sample images.

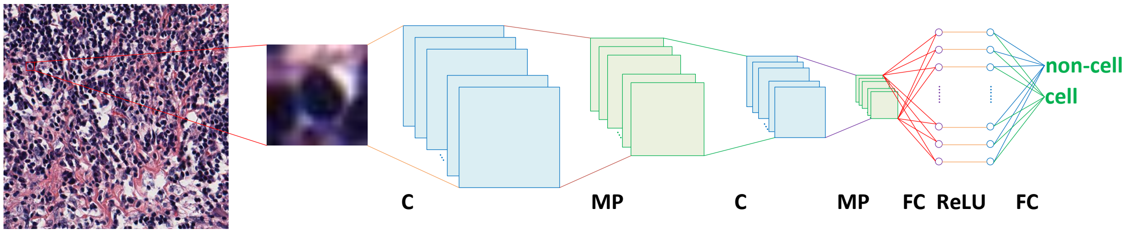


Figure 2.2: The DCNN architecture used in the training process of the proposed framework. C, MP, FC, ReLU represents the convolution layer, max pooling layer, fully connected layer and rectified linear unit layer, respectively.

2.2.2 Deep Convolution Neural Network architecture

Our DCNN model contains two pairs of convolution and max-pooling layers, followed by a fully connected layer, rectified linear unit layer and another fully connected layer as output. Figure 2.2 illustrates the network architecture for training stage. Each **convolution layer** performs a 2D-convolution operation with a square filter. If the activation from previous layer contains more than one map, they are summed up first and then convoluted. In the training process, the stride of **max-pooling layer** is set the same as its kernel size to avoid overlap, provide more non-linearity and reduce dimensionality of previous activation map. The **fully connected layer** mixes the output from previous map into the feature vector. A **rectified linear unit layer** is followed because of its superior non-linearity. The output layer is simply another fully connected layer with just two neurons(one for cell class, the other for non-cell class), activated by a softmax function to provide the final possibility map for the two classes. We detail the layer type, neuron size, filter size and filter number parameters of the proposed DCNN framework in the left of Table 2.2.

Table 2.1: Backward network architecture. M: the number of patch samples, N: the number of testing images. Layer type: I - Input, C - Convolution, MP - Max Pooling, ReLU - Rectified Linear Unit, FC - Fully Connected

Type	Maps and neurons	Filter size	Filter num	Stride
I	$3 \times 20 \times 20M$	-	-	-
C	$20 \times 16 \times 16M$	5	20	1
MP	$20 \times 8 \times 8M$	2	-	2
C	$50 \times 4 \times 4M$	5	50	1
MP	$50 \times 2 \times 2M$	2	-	2
FC	$500M$	1	-	-
ReLU	$500M$	1	-	-
FC	$2M$	1	-	-

2.2.3 Acceleration of Forward Detection

The traditional sliding window manner requires the patch-by-patch scanning for all the pixels in the same image. It sequentially and independently feeds patches to DCNN and the forward propagation is repeated for all the local pixel patches. However, this strategy is time consuming due to the fact that there exists a lot of redundant convolution operations among adjacent patches when computing the sliding-windows.

To reduce the redundant convolution operations, we utilize the relations between adjacent local image patches. In the proposed acceleration model, at the testing stage, the proposed model takes the whole input image as input and can predict the whole label map with just one pass of the accelerated forward propagation. If a DCNN takes $n \times n$ image patches as inputs, a testing image of size $h \times w$ should be padded to size $(h+n-1) \times (w+n-1)$ to keep the size consistency of the patches centered at the boundary of images. The proposed method, in the testing stage, uses the exact weights solved in the training stage to generate the exactly same result as the traditional sliding window method does. To achieve this goal, we involve the k-sparse

Table 2.2: Accelerated forward network architecture. M: the number of patch samples, N: the number of testing images. Layer type: I - Input, C - Convolution, MP - Max Pooling, ReLU - Rectified Linear Unit, FC - Fully Connected

Type	Maps and neurons	Filter size	Filter number	Stride
I	$3 \times 531 \times 531N$	-	-	-
C	$20 \times 527 \times 527N$	5	20	1
MP	$20 \times 526 \times 526N$	2	-	1
C	$50 \times 518 \times 518N$	9	50	1
MP	$50 \times 516 \times 516N$	3	-	1
FC(C)	$500 \times 512 \times 512N$	5	-	1
ReLU	$500 \times 512 \times 512N$	1	-	-
FC(C)	$2 \times 512 \times 512N$	1	-	-

kernel technique[22] for convolution and max-pooling layers into our approach. The k-sparse kernels are created by inserting all-zero rows and columns into the original kernels to make every two original neighboring entries k-pixel away. To accelerate the forward process of fully connect layer, we treat fully connected layer as a special convolution layer. Then the fully connect layer could be accelerated by the modified convolution layer. The proposed fast forwarding network is detailed in Table 3.1(right). Experimental results show that around 400 times speedup is achieved on 512×512 testing images for forward propagation.

2.3 Materials, Experiments and Results

2.3.1 Materials and Experiment Setup

2.3.1.1 Data Set

The proposed method is evaluated on part of the National Lung Screening Trial (NLST) data set [23]. Totally 215 tile images of size 512×512 are selected from the original high-resolution histopathological images. The nuclei in these tiles are

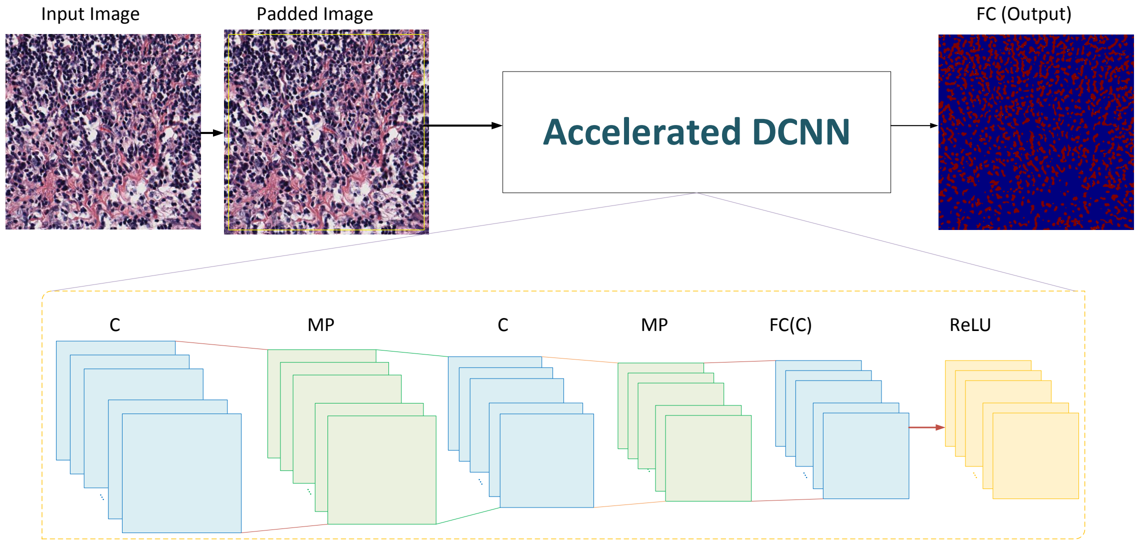


Figure 2.3: The illustration of acceleration forward net: 1) The proposed method takes the whole image as input in testing stage. 2) The input image is mirror padded as the sampling process in the training stage. 3) The padded image is then put into the accelerated forward network which generates the whole label map in the rightmost. Note that the fully connected layer is implemented via a modified convolution layer to achieve acceleration.

manually annotated by the well-trained pathologist. The selected dataset contains a total of 83245 nuclei objects.

2.3.1.2 Experiments Setup

We partition the 215 images into three subsets: training set (143 images), validation set (62 images) and evaluation set (10 images). The evaluation result is reported on evaluation subset containing 10 images. We compare the proposed method with the state-of-the-art method in cell detection[1] and the traditional DCNN-based sliding window method[4].

Table 2.3: F_1 scores on the evaluation set

	1	2	3	4	5	6	7	8	9	10	Mean
MSER[1]	0.714	0.633	0.566	0.676	**0.751**	0.564	0.019	0.453	0.694	0.518	0.559
Proposed	**0.790**	**0.852**	**0.727**	**0.807**	0.732	**0.804**	**0.860**	**0.810**	**0.770**	**0.712**	**0.786**

Table 2.4: Mean time cost comparison on the evaluation set

	1	2	3	4	5	6	7	8	9	10	Mean
MSER[1]	37.897	29.000	37.172	43.332	42.806	37.843	28.548	41.570	38.346	37.012	37.353
Pixel-wise[24]	38.936	38.923	38.306	38.080	37.126	38.038	37.030	37.398	37.407	38.470	37.972
Proposed	**0.128**	**0.124**	**0.116**	**0.115**	**0.114**	**0.125**	**0.115**	**0.127**	**0.116**	**0.126**	**0.121**

2.3.2 Results

2.3.2.0.1 Training Time Cost The mean training time for the proposed method is 229 seconds for the training set described below. The unaccelerated version with the same training strategy costs the same time as the proposed method. Besides, the state-of-the-art MSER-based method[1] costs more than 400000 seconds, roughly 5 days for training 143 images of size 512 × 512. The proposed method is able to impressively reduce several thousand times time cost of training stage than the state-of-the-art MSER-based method due to the proposed training strategy.

2.3.2.1 Accuracy of Testing

Table 2.3 reports the F_1 score metric comparison between the proposed method and MSER-based method. The proposed method outperforms the state-of-the-art method in almost all of the evaluation images in terms of F_1 scores. We also visually compares our results with the MSER-based method in Figure 2.4. The proposed method detects almost all of the cell regions even in images with intensive cells.

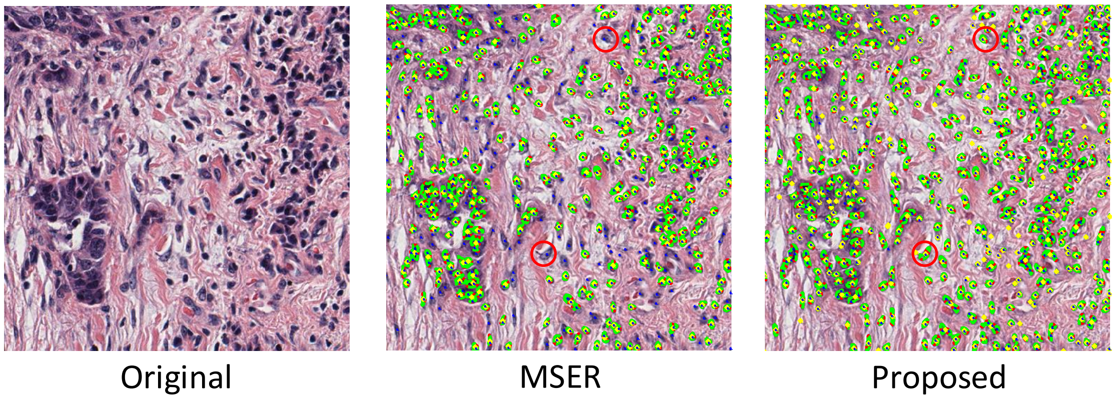


Figure 2.4: Visual Comparison between the proposed method and MSER-based method[1]. The green area denotes the detected cell area by the corresponding method. Blue dots denote the ground-truth annotation. The proposed method is able to detect the cell area missed by the MSER-based method as denoted in red circle. Better viewed in ×4 pdf.

2.3.2.2 Testing Time Cost

As shown in Table 2.4, the proposed method only costs around 0.1 second for a single 512×512 tile image, which is the fastest among the three methods. The proposed method accelerates the forwarding procedure around 400 times compared with the traditional pixel-wise sliding-window method, which is due to the accelerated forwarding technique.

2.4 Conclusion

In this paper, we propose an efficient and robust lung cancer cell detection method. The proposed method is designed based on the Deep Convolution Neural Network framework[24], which is able to provide state-of-the-art accuracy with only weakly annotated ground truth. For each cell area, only one local patch containing the cell area is fed into the detector for training. The training strategy significantly reduces the time cost of training procedure due to the fact that only around one percent of all pixel labels are used. In the testing stage, by utilizing the relation of adjacent patches, the proposed method provides the exact same results within a few hundredths time. Experimental results clearly demonstrate the efficiency and effectiveness of the proposed method for large-scale lung cancer cell detection. In the future, we shall attempt to combine the structured techniques[25, 26, 27] to further improve the accuracy.

CHAPTER 3

DETECTING 10,000 CELLS IN ONE SECOND

In this paper, we present a generalized distributed deep neural network architecture to detect cells in whole-slide high-resolution histopathological images, which usually hold 10^8 to 10^{10} pixels. Our framework can adapt and accelerate any deep convolutional neural network pixel-wise cell detector to perform whole-slide cell detection within a reasonable time limit. We accelerate the convolutional neural network forwarding through a sparse kernel technique, eliminating almost all of the redundant computation among connected patches. Since the disk I/O becomes a bottleneck when the image size scale grows larger, we propose an asynchronous prefetching technique to diminish a large portion of the disk I/O time. An unbalanced distributed sampling strategy is proposed to enhance the scalability and communication efficiency in distributed computing. Blending advantages of the sparse kernel, asynchronous prefetching and distributed sampling techniques, our framework is able to accelerate the conventional convolutional deep learning method by nearly $10,000$ times with same accuracy. Specifically, our method detects cells in a 10^8-pixel ($10^4 \times 10^4$) image in 20 seconds (approximately $10,000$ cells per second) on a single workstation, which is an encouraging result in whole-slide imaging practice. [3].

3.1 Introduction

Recently, increased interests have been raised in the research community concerning the cell detection problem. A large number of cell detection methods on small images (with around 10^4 to 10^6 pixels) have been proposed [1, 28, 29, 30]. Due to the recent success of deep convolutional neural network in imaging, several deep neural network based methods have been proposed for cell-related applications in the past few years [28, 29, 30]. While these methods have achieved great success on small images, very few of them are ready to be applied into practical whole-slide cell detection, in that the real whole-slide images usually have 10^8 to 10^{10} pixels. It takes several weeks to detect cells in a single whole-slide image by directly applying the deep learning cell detection methods [28, 29, 30], which is definitely prohibitive in practice.

To alleviate the issue, we hereby propose a generalized distributed deep convolutional neural network framework for the pixel-wise cell detection. Our framework accelerates any deep convolutional neural network pixel-wise cell detector. In the proposed framework, we first improve the forwarding speed of the deep convolutional neural network with the sparse kernel technique. Similar techniques are referred to [31, 22]. In order to reduce the disk I/O time, we propose a novel asynchronous prefetching technique. The separable iteration behavior also suggests needs for a scalable and communication efficient distributed and parallel computing framework to further accelerate the detection process on whole-slide images. We, therefore, recommend an unbalanced distributed sampling strategy with two spatial dimensions, extending the balanced cutting in [32]. The combination of the aforementioned techniques thus yields a huge speedup up to 10,000x in practice.

To the best of our knowledge, the research presented in this paper represents the first attempt to develop an extremely efficient deep neural network based pixel-wise cell detection framework for whole-slide images. Particularly, it is general enough to cooperate with any deep convolutional neural networks to work on whole-slide imaging. Our technical contributions are summarized as: 1) A general sparse kernel neural network model is applied for the pixel-wise cell detection, accelerating the forwarding procedure of the deep convolutional neural networks. 2) An asynchronous prefetching technique is proposed to reduce nearly 95% of the disk I/O time. 3) We propose a scalable and communication efficient framework to extend our neural network to multi-GPU and cluster environments, dramatically accelerating the entire detecting process. Extensive experiments have been conducted to demonstrate the efficiency and effectiveness of our method.

3.2 Methodology

3.2.1 Sparse Kernel Convolutional Neural Network

The sparse kernel network takes the whole tile image, instead of a pixel-centered patch, as input and can predict the whole label map with just one pass of the accelerated forward propagation. The sparse kernel network uses the same weights as the original network trained in the training stage to generate the exact same results as the original pixel-wise detector does. To achieve this goal, we involve the k-sparse kernel technique [22] for convolution and blended max-pooling layers into our approach. The k-sparse kernels are created by inserting all-zero rows and columns into the original kernels to make every two original neighboring entries k-pixel away. In [22], however, it remains unclear how to deal with fully connected layers, which is completed in our research. A fully connected layer is treated as a special convolution layer with kernel size set to the input dimension and kernel number set as the output dimension of the fully connected layer. This special convolution layer will generate the exact same output as the fully connected layer does when given the same input. The conversion algorithm is summarized in Algorithm 1.

3.2.2 Asynchronous Prefetching

Comparing with other procedures in the whole cell detection process, e.g. the memory transfer between GPU and CPU memory, the disk I/O becomes a bottleneck in the cell detection problem. In this subsection, we describe our asynchronous prefetching technique to relieve the bottleneck of the disk I/O. To reduce frequent I/O operations and, meanwhile, ensure the absence of insufficient memory problems, we propose an asynchronous prefetching technique to resolve this. We first load a relatively large image, referred to as *cached image*, into memory (e.g., 4096×4096). While we start to detect cells on the first cached image tile by tile, we immediately start loading the second cached image in another thread. Thus, when the detection process of the first cached image is finished, since the reading procedure is usually faster than the detection, we've already loaded the second cached image and can start detection in the second cached image and load the next cached image immediately. Hence, the reading time of the second cached image, as well as the cached images thereafter, is hidden from the overall runtime. Experiments have exhibited that this technique reduces approximately 95% of the disk I/O time. It achieves an even larger speedup on a cluster since the NFS (Network File System) operation is even more time-consuming and we reduce most of them.

3.2.3 Multi-GPU Parallel and Distributed Computing

When considering distributed optimization, two resources are at play: 1) the amount of processing on each machine, and 2) the communication between machines. The single machine performance has been optimized in Section 4.2.3.0.3 and 3.2.2. We then describe our unbalanced distributed sampling strategy with two spatial dimensions of our framework, which is a gentle extension to [32]. Assuming

$T = \{(1,1), (1,2), \ldots, (H,W)\}$ is the index set of an image with size $H \times W$, we aim at sampling tiles of sizes not larger than $h \times w$.

3.2.3.0.1 Unbalanced Partitioning Let $S := \lceil HW/C \rceil$. We first partition the index set T into a set of blocks $P^{(1)}, P^{(2)}, \ldots, P^{(C)}$ according to the following criterion:

1. $T = \bigcup_{c=1}^{C} P^{(c)}$,
2. $P^{(c')} \bigcap P^{(c'')} = \varnothing$, for $c' \neq c''$,
3. $|P^{c}| \leq S$,
4. $P^{(c)}$ is connected.

3.2.3.0.2 Sampling After the procedure of partitioning, we now sample small tiles from C different machines and devices. For each $c \in \{1, \ldots, C\}$, the $\hat{Z}^{(c)}$ is a connected subset of $P^{(c)}$ satisfying $|\hat{Z}^{(c)}| \leq hw$ and $\hat{Z}^{(c')} \bigcap \hat{Z}^{(c'')} = \varnothing$, for $c' \neq c''$.

The set-valued mapping $\hat{Z} = \bigcup_{c=1}^{C} \hat{Z}^{(c)}$ is termed as (C, hw)-unbalanced sampling, which is used for fully sampling tile images from the entire image. Note this is not a subsampling process since all the tile images are sampled from the whole slide in one data pass. Since only index sets are transmitted among all the machines, the communication cost is very low in network transferring. This distributed sampling strategy also ensures the scalability of the proposed framework as indicated in Section 3.3.4.

3.3 Experiments

3.3.1 Experiment Setup

Throughout the experiment section, we use a variant [30, 2][1] of LeNet [4] as a pixel-wise classifier to show the effectiveness and efficiency of our framework. We have implemented our framework based on caffe [24] and MPI. The original network structure is shown in Table 3.2 (left). The classifier is designed to classify a 20 × 20 patch centered at specific pixel and predict the possibility of whether the pixel is in a cell region. Applying Algorithm 1, we show the accelerated network on the right of Table 3.2, which detects cells on a tile image of size 512 × 512. Since the classifier deals with 20 × 20 image patches, we mirror pad the original 512 × 512 tile image to a 531 × 531 image.

Table 3.1: Original LeNet Classifier network architecture. M: the training batch size, N: the testing batch size. Layer type: I - Input, C - Convolution, MP - Max Pooling, ReLU - Rectified Linear Unit, FC - Fully Connected

Type	Maps and neurons	Filter size	Filter num	Stride
I	$3 \times 20 \times 20M$	-	-	-
C	$20 \times 16 \times 16M$	5	20	1
MP	$20 \times 8 \times 8M$	2	-	2
C	$50 \times 4 \times 4M$	5	50	1
MP	$50 \times 2 \times 2M$	2	-	2
FC	$500M$	1	-	-
ReLU	$500M$	1	-	-
FC	$2M$	1	-	-

[1] The code is the publicly available at `https://github.com/uta-smile/caffe-fastfpbp`. We also provide a web demo for our method at `https://celldetection.zhengxu.work/`.

Table 3.2: Accelerated forward network architecture. M: the training batch size, N: the testing batch size. Layer type: I - Input, C - Convolution, MP - Max Pooling, ReLU - Rectified Linear Unit, FC - Fully Connected

Type	Maps and neurons	Filter size	Filter number	Stride
I	$3 \times 531 \times 531N$	-	-	-
C	$20 \times 527 \times 527N$	5	20	1
MP	$20 \times 526 \times 526N$	2	-	1
C	$50 \times 518 \times 518N$	9	50	1
MP	$50 \times 516 \times 516N$	3	-	1
FC(C)	$500 \times 512 \times 512N$	5	-	1
ReLU	$500 \times 512 \times 512N$	1	-	-
FC(C)	$2 \times 512 \times 512N$	1	-	-

3.3.2 Effectiveness Validation

Our framework can be applied to any convolutional neural network for pixel-wise cell detection, e.g., [28, 29, 30]. Thus, the effectiveness of our framework highly depends on the performance of the original deep neural networks designed for the small-scale cell detection. In this subsection, we validate the result consistency between our framework and the original work [30]. We conduct experiments on 215 tile images sized 512 × 512 sampled from the NLST[2] whole-slide images, with 83245 cell object annotations. These tile images are then partitioned into three subsets: the training set (143 images), the testing set (62 images) and the evaluation set (10 images). The neural network model was trained on the training set with the original network described on the Table 3.2 (left). We then applied Algorithm 1 to transfer the original network into our framework. This experiment was conducted on a workstation with Intel(R) Xeon(R) CPU E5-2620 v2 @ 2.10GHz CPU, 32 gigabyte RAM, and a single Nvidia K40 GPU.

[2] https://biometry.nci.nih.gov/cdas/studies/nlst/

For quantitative analysis, we used a precision-recall-F_1score evaluation metric to measure the performance of the two methods. Since the proposed method detects the rough cell area, we calculated the **raw image moment** centroid as its approximate nuclei location. Each detected cell centroid is associated with the nearest ground-truth annotation. A detected cell centroid is considered to be a True Positive (TP) sample if the Euclidean distance between the detected cell centroid and the ground-truth annotation is less than 8 pixels; otherwise, it is considered as False Positive (FP). Missed ground-truth dots are counted as False Negative (FN) samples. We consider F_1 score $F_1 = 2PR/(P+R)$, where precision $P = TP/(TP+FP)$ and recall $R = TP/(TP + FN)$. We report the precision, recall and F_1 score of the original work and our framework in Table 3.3.

Table 3.3: Quantitative Comparison between Original Work and Our Framework

Methods	Precision	Recall	F_1 score	Overall Runtime	Pixel Rate
Original Work[30]	0.83±0.09	0.84±0.10	0.83±0.07	38.47±1.01	6814.24±174.43
Our Framework	0.83±0.09	0.84±0.10	0.83±0.07	**0.10±0.00**	**2621440.00±24.01**

Table 3.3 also shows the overall runtime (in seconds) and pixel rate (pixels per second) comparison. While our framework produced the same result as the original work, our overall speed was increased by approximately 400 times in small scale images on a single GPU device. This is reasonable since our method reduces most redundant convolution computation among the neighbor pixel patches.

3.3.3 Prefetching Speedup

In this subsection, we validate the effectiveness of the proposed asynchronous prefetching technique. Fig.3.1 shows the disk I/O time comparison among memory, file and prefetching modes in a whole-slide image (NLSI0000105 with spatial dimension 13483 × 17943). The I/O time is calculated by the difference between the overall runtime and the true detection time. As mentioned in Section 3.2.2, memory mode is slightly faster than file mode in that memory mode requires less hardware interruption invocation. Note that the prefetching technique doesn't truly reduce the I/O time. It hides most I/O time into the detection time, since the caching procedure and detection occur simultaneously. So for a 10^8-pixel whole-slide image, our technique diminishes (or hides) 95% I/O time compared with file mode. This is because the exposed I/O time with our prefetching technique is only for reading the first cached image.

3.3.4 Parallel and Distributed Computing

In this subsection, we show our experiment results in several whole-slide images. We randomly selected five whole-slide images, in Aperio SVS format, from NLST and TCGA [33] data sets, varying in size, from 10^8 to 10^{10} pixels. In order to show the efficiency of our methods, we conducted experiments in all five whole-slide images on a single workstation with Intel(R) Core(TM) i7-5930K CPU @ 3.50GHz, 64 Gigabytes RAM, 1 TB Samsung(R) 950 Pro Solid-State Drive and four Nvidia Titan X GPUs. Table 3.4 shows the overall runtime on cell detection in these whole-slide images. On a single workstation, our method is able to detect cells in a whole-slide image of size around $10^4 \times 10^4$ (NLSI0000105) in 20 seconds. Since the detection result of this whole-slide image includes

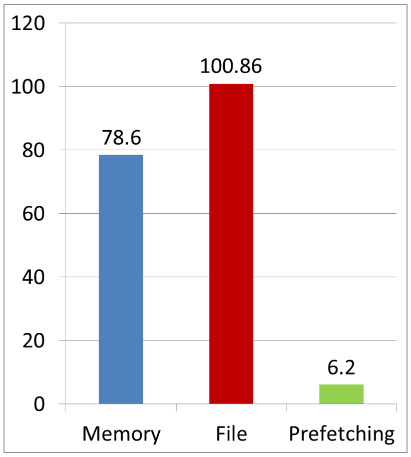


Figure 3.1: I/O Time Comparison among Memory, File and proposed Asynchronous Prefetching modes (in seconds)

approximately $200,000$ cells, our method detects nearly $10,000$ cells per second on average on a single workstation, while the original work [30] only detects approximately 6 cells per second, reaching a $1,500$ times speedup.

Table 3.4: Time Comparison on Single Workstation (in seconds)

Image Name (Dimension)	1 GPU	2 GPUs	3 GPUs	4 GPUs
NLSI0000105 (13483 × 17943)	71.43	38.81	26.89	20.88
NLSI0000081 (34987 × 37879)	366.74	194.99	131.30	99.20
TCGA-05-4405 (83712 × 50432)	1502.16	800.24	529.00	449.94
TCGA-35-3615 (62615 × 133335)	2953.99	1519.57	1100.32	861.17
TCGA-38-4627 (65033 × 149642)	3385.28	1773.11	1216.80	972.36

The workaround of our method in distributed computing environment is demonstrated on TACC Stampede GPU clusters[3]. Each node is equipped with two 8-core Intel Xeon E5-2680 2.7GHz CPUs, 32 Gigabytes RAM and a single Nvidia K20 GPU. We show only the distributed results for the last four images from Table 3.4, since the first image is too small to be sliced into 32 pieces. Table 3.5 shows that our method detects cells in a whole-slide image (TCGA-38-4627) with nearly 10^{10} pixels within 155.87 seconds. When directly applying the original work, it takes approximately 400 hours (1440000 seconds) even without considering the disk I/O time. Our method has impressively achieved nearly $10,000$ times speed up compared with naively applying [30]. The linear speedup also exhibits the scalability and communication efficiency, since our our sampling strategy reduces most overhead in communication.

3.4 Conclusions

In this paper, a generalized distributed deep neural network framework is introduced to detect cells in whole-slide histopathological images. The innovative frame-

[3] `https://www.tacc.utexas.edu/stampede/`

Algorithm 1 Network To Sparse Kernel Network Conversion Algorithm

```
Input: Original network $\mathcal{N}$ with $K$ layers denoted as $\mathcal{N} = \{\mathcal{N}^{(1)}, \ldots, \mathcal{N}^{(K)}\}$.
Output: Sparse kernel network $\hat{\mathcal{N}}$ with $K$ layers.
Initialization: $d = 1$
for $k = \{1, 2, \ldots, K\}$ do
    if $\mathcal{N}^{(k)}$ is convolution layer then
        Set $\hat{\mathcal{N}}^{(k)}$ as ConvolutionSK layer
        $\hat{\mathcal{N}}^{(k)}_{stride} := 1$, $\hat{\mathcal{N}}^{(k)}_{kstride} := d$, $\hat{\mathcal{N}}^{(k)}_{kernel} := \mathcal{N}^{(k)}_{kernel}$
    else if $\mathcal{N}^{(k)}$ is pooling layer then
        Set $\hat{\mathcal{N}}^{(k)}$ as PoolingSK layer
        $\hat{\mathcal{N}}^{(k)}_{stride} := 1$, $\hat{\mathcal{N}}^{(k)}_{kstride} := d$
    else if $\mathcal{N}^{(k)}$ is fully connected layer then
        Set $\hat{\mathcal{N}}^{(k)}$ as ConvolutionSK layer
        $\hat{\mathcal{N}}^{(k)}_{stride} := 1$, $\hat{\mathcal{N}}^{(k)}_{kstride} := d$, $\hat{\mathcal{N}}^{(k)}_{num\_output} := \mathcal{N}^{(k)}_{num\_output}$
        $\hat{\mathcal{N}}^{(k)}_{kernel\_size} := \mathcal{N}^{(k-1)}_{output\_shape}$, $\hat{\mathcal{N}}^{(k)}_{kernel} := \mathcal{N}^{(k)}_{weight}$
    else
        $\hat{\mathcal{N}}^{(k)} = \mathcal{N}^{(k)}$
    end if
    $d := d \times \mathcal{N}^{(k)}_{stride}$
end for
```

Table 3.5: Time Comparison on Multi-node Cluster (in seconds)

Image Name (Dimension)	1	2	4	8	16	32
NLSI0000081 (34987 × 37879)	520.94	266.06	143.99	77.16	44.10	26.03
TCGA-05-4405 (83712 × 50432)	1820.08	945.77	508.23	271.02	155.39	86.31
TCGA-35-3615 (62615 × 133335)	3558.48	1834.00	944.91	487.47	266.35	147.07
TCGA-38-4627 (65033 × 149642)	4151.56	2107.46	1086.53	559.28	293.98	155.87

work can be applied with any deep convolutional neural network pixel-wise cell detector. Our method is extremely optimized in distributed environment to detect cells in whole-slide images. We utilize a sparse kernel neural network forwarding technique to reduce nearly all redundant convolution computations. An asynchronous prefetching technique is recommended to diminish most disk I/O time when loading the large histopathological images into memory. Furthermore, an unbalanced distributed sampling strategy is presented to enhance the scalability and communication efficiency of our framework. These techniques construct three pillars of our framework. Extensive experiments demonstrate that our method can approximately detect $10,000$ cells per second on a single workstation, which is encouraging for high-throughput cell data. While our result enables the high speed cell detection, our result can expect to benefit some further pathological analysis, e.g. feature extraction [34].

CHAPTER 4

SEQ2SEQ FINGERPRINT: AN UNSUPERVISED DEEP MOLECULAR EMBEDDING FOR DRUG DISCOVERY

Many of today's drug discoveries require expertise knowledge and insanely expensive biological experiments for identifying the chemical molecular properties. However, despite the growing interests of using supervised machine learning algorithms to automatically identify those chemical molecular properties, there is little advancement of the performance and accuracy due to the limited amount of training data.

In this paper, we propose a novel unsupervised molecular embedding method, providing a continuous feature vector for each molecule to perform further tasks, e.g., solubility classification. In the proposed method, a multi-layered Gated Recurrent Unit (GRU) network is used to map the input molecule into a continuous feature vector of fixed dimensionality, and then another deep GRU network is employed to decode the continuous vector back to the original molecule. As a result, the continuous encoding vector is expected to contain rigorous and enough information to recover the original molecule and predict its chemical properties. The proposed embedding method could utilize almost unlimited molecule data for the training phase. With sufficient information encoded in the vector, the proposed method is also robust and task-insensitive. The performance and robustness are confirmed and interpreted in our extensive experiments [10].

4.1 Introduction

In the most recent decade, every drug company with R&D department has carried out numerous initiatives for speeding up its drug discovery process [35]. Drug discovery is the process through which potential new medicines are identified. Modern drug discovery is usually implemented as drug compound selection, while, for every candidate chemical compound, the chemical drug properties, e.g., affinity, selectivity, metabolic stability, are biologically tested in the lab environment. Once all the properties pass the drug requirement tests, it will be selected as a new potential drug candidate. However, this process is excessively expensive and labor-intensive, and costs hundreds of million dollars each year.

Therefore, using machine learning methods to automatically predict the chemical properties has recently raised great interests in the drug discovery community [36, 37, 38, 39]. However, the majority of machine learning algorithms take fixed-length continuous feature vectors as inputs [40, 41, 42]. However, the nature of molecules makes it extremely hard to represent molecules with fixed-length vectors [43, 44, 45]. The readers might refer to Figure 5.1 to grab some intuition. As a simple example, we may consider H_2O (water) and O_2 (oxygen). They differ in atom types, numbers as well as bond types. One might find it is tricky to represent each molecule as a fixed-length vector. So a large class of research papers has been published to generate the fixed-length continuous vector representation for molecules. Overall, the choice of the representation of molecules is at the heart of the machine learning-based drug discovery [46, 47, 48, 49].

Traditionally, the design of new fixed-length vector molecular representations, named **fingerprints**, is not data-driven and based on human expertise knowledge [59, 60, 61, 62, 63]. One type of those design is based on some hashing procedure, e.g., Extended Connectivity FingerPrint (ECFP) [56]. Those fingerprints are usually

Table 4.1: Comparison among different types of fingerprint methods, in three different aspects: 1) if the design of the fingerprint requires biologists' expertise knowledge, 2) if the fingerprint has enough information to be reverted to original SMILE representation, and 3) if the fingerprint method requires many labeled data. DL is short for Deep learning while FP is short for FingerPrint.

Properties	Non-data driven Methods		Supervised DL FP [50, 51, 52]	Seq2seq FP (**ours**)
	Hash-based [53, 54, 55, 56]	Local feature[57, 58]		
Without biologist guide	✓	×	✓	✓
Revertible	×	×	×	✓
Less thirsty on label data	✓	✓	×	✓

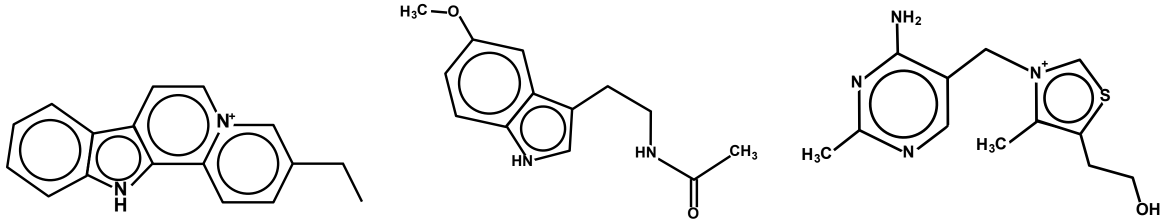
efficient in speed, but is much like a lossy compression in the imaging area [64, 65, 66, 67] and the operation is non-invertible. The other sort of non-data-driven fingerprint is based on local sub-structures of molecules. Biologists look for several highly related chemical molecular sub-structures for specific tasks and design the fingerprint feature vector accordingly. Representative works are [57, 58]. However, this kind of design obviously requires years of expertise experience and is highly task-sensitive. To sum up, the non-data-driven fingerprint is either limited in encoding enough information or highly lean to expensive and accurate human knowledge. Hence it has raised a great demand for the data-driven fingerprints, which does not require years of human guide and expensive biological experiments.

Observed the recent success of deep learning on imaging understanding [68, 2] or natural language processing [69, 70], there are a few attempts made in applying deep neural network to generate fingerprints. Among the most famous ones are the neural fingerprint [52] as well as [50, 51, 71, 72]. However, most supervised deep learning methods are data-hungry and usually completely fail when data scale is limited [73, 74], and unfortunately this is usually the case in the drug discovery due to the insane expensiveness of the lab experiment.

In this paper, we propose an unsupervised data-driven deep-learning-based molecular fingerprint method, named **seq2seq fingerprint**. To overcome the issues mentioned above, 1) the proposed method is data-driven, without any human expertise knowledge required. 2) the fingerprints generated by the proposed method are completely revertible to original molecular representations, ensuring the sufficiency of information encoded in the fingerprint vector. 3) the proposed method employs an unsupervised training on a **huge unlabeled** dataset, sufficiently releasing the horsepower of deep neural network. We illustrate a comparison among all mentioned fingerprint methods and our seq2seq fingerprint method.

Our fingerprint is designed based on a recent breakthrough model, called *sequence-to-sequence learning* (seq2seq learning). The seq2seq learning method comes from a seemingly unrelated area, English-to-French translation. The seq2seq learning method takes an English sentence as the input, encodes it into a *meaning* vector and then translates it back to a French sentence as the output. The crux of our method is similar, but differs in the way that we set both the input and output of the seq2seq learning as the same SMILE string, a text representation of a molecule. We map the SMILE string to a fixed-sized vector and then *translates* it back to the original SMILE string. The intermediate fixed-sized vector is extracted as the **seq2seq fingerprint**. Once the model is well-trained, the intermediate feature vector is considered to encode all the information to recover the original molecular representation. Hence, the seq2seq fingerprint is expected to capture the rigorous information with which we can accurately predict the molecular properties.

The benefits of the seq2seq fingerprint are three folds: 1) the training phase of seq2seq fingerprint is completely **label-free**, avoiding the costly and labor-intensive label acquiring procedure. 2) it is data-driven, eliminating the reliance on expert's subjective knowledge. 3) since the unlabeled data is almost unlimited in practice, we can fully utilize the power of deep learning, without suffering from the short supply of labeled data.



(a) Flavopereirin (b) Melatonin (c) Thiamine

Figure 4.1: The examples of SMILE representations.

The technical contributions of this paper are summarized as: 1) the seq2seq fingerprint method is clearly the first attempt to apply the seq2seq learning method to perform drug discovery tasks, coupling two seemingly unrelated areas. 2) several important adaptations are made into the original seq2seq learning to suit drug discovery applications:

- GRU cell is used, instead of LSTM, to accelerate the training process,
- Attention Mechanism is employed to centralize the fingerprint space,
- Dropout layer is added to overcome the over-fitting issue during the training phase,
- An extra fingerprint extraction layer set is added to pull the fingerprint out.

3) extensive experiments confirm the superior performance on different tasks over the state-of-the-art methods.

The rest of the paper is organized as follows. We summarize several related work, in both drug discovery and sequence to sequence learning, in Section 5.2. In Section 5.3, we describe our entire pipeline in details. We show our experiment results in Section 5.4, demonstrating the superior performance of our method. We conclude and discuss the future direction of our paper in Section 5.5.

4.2 Related Work

In this section, we present several related work. First, we introduce the initial representation of molecules, i.e., how the molecular data is persisted in the data store. Second, we list several state-of-the-art fingerprint methods, including the most recent ones using deep learning techniques. Last but not least, we briefly describe our cornerstone learning method, i.e., seq2seq learning, with several of its related work in language translation area.

4.2.1 SMILE Representations of Molecules

Initially, the molecules are represented through the Simplified Molecular-Input Line-Entry system (SMILE) [75], which is a line notation for describing the structure of chemical species using text strings. The SMILE system represents the chemical structures in a graph-based definition, where the atoms, bonds and rings are encoded in a graph. Simple examples of SMILE representations are 1) dinitrogen with structure $N \equiv N$ (N#N), 2) methyl isocyanate with structure $CH_3 - N = C = O$ (CN=C=O), where corresponding SMILE representations are included in the brackets. We show some more complex examples in Figure 5.1.

4.2.2 Fingerprint Methods

4.2.2.0.1 Hash-based Fingerprints Many hash-based has been developed to generate unique molecular feature representation [53, 54, 55]. One of the most famous ones being Extended-Connectivity FingerPrint (ECFP) [56]. Circular fingerprints generate each layer's features by applying a fixed hash function to the concatenated features of the neighborhood in the previous layer. However, due to the non-invertible nature of the hash function, the hash-bashed fingerprint methods usually do not encode enough information and hence result in lower performance in the further predictive tasks.

4.2.2.0.2 Biologist-guided Local-Feature Fingerprints Another mainstream of traditional fingerprint methods is designed based on the biological experiments and the expertise knowledge and experience, [57, 58]. Biologist look for several important task-related sub-structures (fragments), e.g., $CC(OH)CC$ for pro-solubility prediction, and count those sub-structures as local features to produce fingerprints. This kind of fingerprint methods usually work well for specific tasks, but generalize very poorly for other tasks.

4.2.2.0.3 Supervised Deep Learning-based Fingerprints The growth of deep learning has provided the flexibility and performance to create the molecular fingerprint from data samples, without explicit human guide, [52, 50, 51, 76, 77]. The state-of-the-art work is the neural fingerprint [52]. The neural fingerprint mimics the process of generating circular fingerprint but instead the hash function is replaced by a non-linear activated densely connected layer. This method is based on the data-hungry deep neural network. To acquire enough labeled data, biologists need to perform a sufficiently large number of tests on chemical molecules, which is extremely expensive.

4.2.3 Encoder-Decoder Structured Neural Network

4.2.3.0.1 Variational Auto-Encoder Variational Auto-Encoder (VAE) model [78] shares some similar structure as our method, which uses a *encoder* to encode the original representation to a vector or scalar then a *decoder* to decode the vector to original representation. The difference is that the VAE model puts the assumption that the embedded space follows some specific Gaussian distribution. In most recent months, the authors become aware of an unpublished VAE report in drug discovery [79]. However, there is no evidence and experimental results to support the Gaussian assumption on the embedded fingerprint space. Moreover, we still lack the evaluation on how the VAE will perform in the predictive tasks.

4.2.3.0.2 Generative Adversarial Network Generative Adversarial Network (GAN) [80] has recently become popular in the machine learning area. A GAN is constructed by a *discriminator* and a *generator*. The discriminator acts as a cop to distinguish the training data samples from the samples generated from the generator. Hence, the learning process actually learns from both training data set and the generated fake data samples. It works well when the scale of data sample is limited. But such network is hard to train and we are not aware of any publicly available report that documents the attempt to adapt GAN into drug discovery.

4.2.3.0.3 Sequence to Sequence Model The sequence to sequence model [81] has been recently used in English-to-French translation and demonstrated as a breakthrough success. The basic strategy of sequence to sequence learning is to map the input sequence, e.g., an English sentence, to a fixed-sized vector using one deep Long Short-Term Memory (LSTM) network, and then map the vector to the target sequence, e.g., the translated French sentence, with another deep LSTM network. The fixed-sized vector is considered as an intermediate representation and contains the *"meaning"* of sentences.

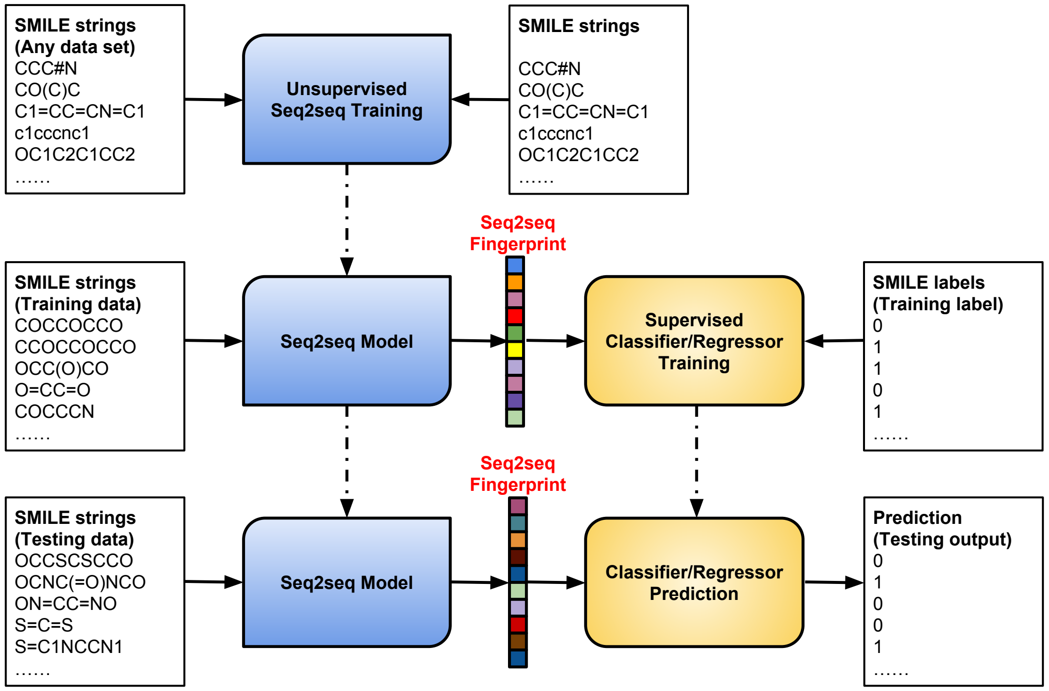


Figure 4.2: This figure shows how the entire pipeline works. 1) The seq2seq fingerprint model is trained on a large pool of unlabeled SMILE data. 2) The trained model is fed with SMILE strings to generate the seq2seq fingerprint. 3) Coupling the fingerprint and label, the pairs are fed into supervised classifiers/regressors to train a predictive model.

4.3 Methodology

In the sequel, we present the entire pipeline of the proposed method. First, we overview the entire pipeline with an introduction of the crux of our fingerprint method. Second, we detail each step of our method and our improvements and adaptations upon the original seq2seq learning method. Last, we discuss our methods to end this section.

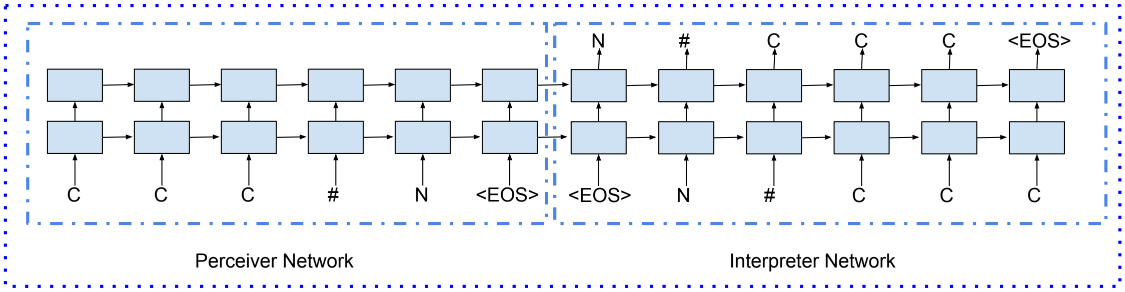


Figure 4.3: An example on how the unsupervised training works. The **perceiver network** understands the molecule SMILE representation, e.g., CCC#N, and encodes it into a fixed-length vector, i.e., **seq2seq fingerprint**. The **interpreter network** will then translate the fingerprint back to the sequence, e.g. N#CCC.

4.3.1 Overview

The entire pipeline of our method consists of three steps: 1) we first train the seq2seq fingerprint model on a **huge** pool of unlabeled training data. 2) Then the trained model is used to generate the seq2seq fingerprint for the labeled data set. 3) The resulting fingerprints and their labels are fed to some supervised learning method to train a predictive model, e.g., Gradient Boosting, Multi-Layer Perceptron (MLP). An illustration of the pipeline is shown in Figure 5.3. As a result, the entire pipeline is able to **transfer** knowledge from a large number of unlabeled data samples to the supervised training on a relatively small labeled dataset and thus improve the final predictive performance.

The crux of our unsupervised seq2seq fingerprint method, **seq2seq fingerprint**, is to set both input and output sequences to the same SMILE string for each molecule in sequence-to-sequence learning for unsupervised training, or simply **translate the SMILE string to itself**. Since the intermediate vector is considered to maintain the "meaning" of the sequence, we thus extract the intermediate vector as the fingerprint. While the sequence to sequence learning [81] could in principle directly work with our idea, however, there are still many drawbacks yet to limit the application in the molecular predictive tasks.

First, at least in theory, our model can train on a pool of infinite molecular data. While the LSTM is famous for its slow training, the time invested in the training process on a large amount of data is absolutely unbearable. Second, the original sequence to sequence learning does not explicitly output the embedding vector, and therefore lacks an extra layer in order to output the fingerprint vector. Third, as argued in [82], when the length of the input sequence grows, the performance of the neural network decreases rapidly. However, SMILE representation usually contains several tens of characters (up to 250 characters), which is too long to be handled by the original sequence to sequence model. Finally, due to the large training data scale, the number of model parameters tends to be relatively smaller than demanded, yielding the over-fitting issue.

Here, we propose the seq2seq fingerprint, with various of improvement upon the original sequence-to-sequence learning [81] used in English-to-French translation to generate an effective fingerprint for drug discovery tasks. We detail each step in the following sub-sections.

4.3.2 Unsupervised Seq2seq Training

To train a fingerprint generator on a huge unlabeled dataset, we first employ a deep Gated Recurrent Unit (GRU) network, named *perceiver network*, to map the original molecular SMILE string to a fixed-sized vector, i.e., the seq2seq fingerprint. Then another deep GRU neural network, called *interpreter network*, is used to generate the original SMILE string back from the seq2seq fingerprint. A work-flow illustration of our method is shown in Figure 4.3. In the following, we show, in the descent order of importance from the authors' perspective, the details we altered to adapt the drug discovery application.

GRU Units The Gated Recurrent Unit (GRU) is used in our experiment instead of LSTM. GRU is famous for its LSTM-like performance but faster training process. A GRU network computes a sequence of outputs $(s_1, \ldots, s_T)$ from the input sequences $(x_1, \ldots, x_T)$ by iterating

$$\begin{aligned} z_t &= \sigma_g(W_z x_t + U_z s_{t-1} + b_z) \\ r_t &= \sigma_r(W_r x_t + U_r s_{t-1} + b_r) \\ h_t &= \tanh(U_h x_t + W_h(s_{t-1} \circ r_t)) \\ s_t &= (1 - z_t) \circ h_{t-1} + z_t \circ s_{t-1}. \end{aligned} \tag{4.1}$$

A GRU cell has two gates: the update gate z and the reset gate r. Each gate has the trainable parameter W, U, b. The activation function σ for each gate is usually the sigmoid function. GRU also has the *"hidden memory"* h, which holds another set of trainable parameters U, W. In contrast with LSTM which has three gates, it has similar performance but faster training speed [83].

Attention Mechanism So far, the only connection between the perceiver and interpreter networks is the sharing hidden memory. When the sequence becomes longer, it becomes extremely challenging to pass the information from the perceiver to the interpreter network through the hidden memory. To address this issue, the attention mechanism is employed to establish a stronger connection and provide soft-alignment between the perceiver and interpreter networks. More details are referred to [82].

Dropout Layer One of the most favorable features in our model is the capability to use nearly unlimited molecular training data. However, the over-fitting issue will come to play if we grow our data unrestrictively. To enhance the generalizability of our model, we add dropout layer to each input, output gate and yet we do not add the dropout for the hidden memory transferring gate, following the practices in [84].

What do we inherit? While we improve the original sequence-to-sequence model from several aspects, we keep using the reverse technique introduced in [81], where the source sequence is mapped to the reverse sequence of the target. For example, instead of mapping a, b, c to α, β, γ, we map a, b, c to γ, β, α. This trick is observed to make it easier for the Stochastic Gradient Descent (SGD) algorithm to *"establish communication"* between the source and target sequences. Another important technique we keep is the *bucket training*, where all the training sequences are distributed into several buckets, and all the sequences in the same bucket are padded to the same length. This technique can parallel the training process on GPUs for acceleration.

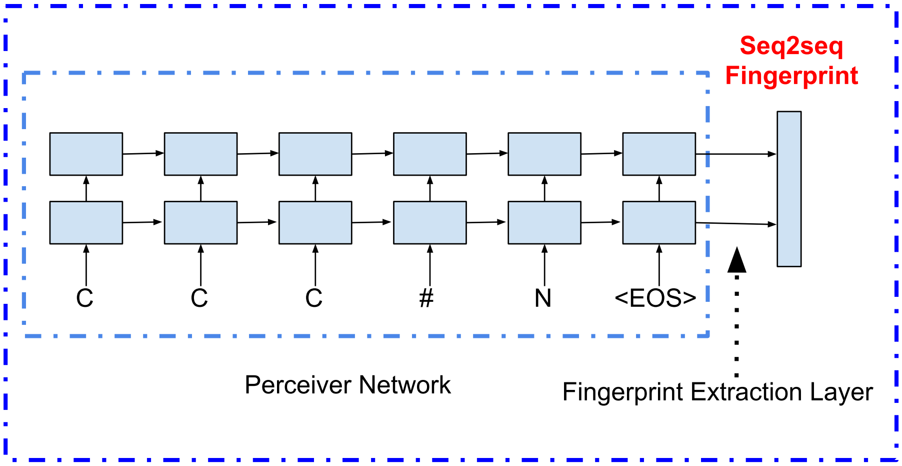


Figure 4.4: The illustration of how to extract the seq2seq fingerprint. Only the perceiver network is feed-forwarded with an extra fingerprint extraction layer to extract the resulting seq2seq fingerprint.

4.3.3 Fingerprint Extraction

During the fingerprint extraction stage, we only feed-forward the *perceiver* network, leaving the interpreter network behind to save computational resources. Moreover, the original sequence to sequence model does not explicitly output the embedding vector, which brings us challenges to extract fingerprints we need. A fixed unit fully connected layer together with a GRU cell state concatenation layer is injected between the perceiver network and interpreter network to extract the seq2seq fingerprint from the network. The illustration of this process is in Figure 4.4.

4.3.4 Supervised Training on Labeled Data

Since our method embedded the molecular graph into a vector space with fixed dimension, the resulting fingerprint can be almost trained with almost all popular regressors or classifiers. Those methods include but not limited to linear Support Vector Machine (SVM), ν-support vector machine, and ensemble methods, e.g., AdaBoost, Extra Trees, etc. In our experiments, we investigate our fingerprints with three ensemble methods: AdaBoost, GradientBoost and Random Forest.

4.3.5 Discussion

Our method can indeed transfer knowledge from unlabeled data to the labeled data training. However, it is not technically semi-supervised, since the unlabeled data is not directly used in the supervised training. So we still name our fingerprint method as unsupervised.

4.4 Experiments

In this section, we first detail the experimental setup, e.g., the data set description, hardware and software settings, etc. Then we report the recovery performance

Table 4.2: The comparison of classification accuracy on the LogP data.

			Adaboost (Ours)			GradientBoost (Ours)			RandomForest (Ours)		
	Circular	Neural	512	768	1024	512	768	1024	512	768	1024
Mean	0.3674	0.6080	0.7044	0.6837	0.7342	0.7350	0.7149	**0.7664**	0.6895	0.6664	0.6845
StDev	0.0074	0.0135	0.0042	0.0097	0.0042	0.0060	0.0058	0.0043	0.0061	0.0100	0.0032

Table 4.3: The comparison of classification accuracy on the PM2-10k data.

			Adaboost (Ours)			GradientBoost (Ours)			RandomForest (Ours)		
	Circular	Neural	512	768	1024	512	768	1024	512	768	1024
Mean	0.3938	0.5227	0.5535	0.5561	0.6036	0.5741	0.5713	**0.6206**	0.5316	0.5282	0.5481
StDev	0.0114	0.0112	0.0132	0.0070	0.0147	0.0086	0.0151	0.0198	0.0110	0.0081	0.0088

of the seq2seq fingerprint method, i.e., how the SMILE self-translation performance is. Finally, we show the superior performance on two predictive tasks for our seq2seq fingerprint method.

4.4.1 Experiment Setup

Unsupervised Train Dataset Our training data was collected from a combination of two large datasets: LogP and PM2-full datasets. Those datasets were obtained from National Center for Advancing Translational Sciences (NCATS) at National Institutes of Health (NIH). The training dataset contains 334,092 valid molecular SMILE representations.

Labeled Datasets We performed the classification on two smaller datasets:

- **LogP**: LogP dataset contains a total of 10,850 samples. Each sample contains a pair of a SMILE string and a water-octanol partition coefficient (LogP) value. A certain threshold of 1.88 is suggested by an NCATS expert. Samples with LogP value smaller than 1.88 will be classified as the negative samples, while the opposites are considered the positive samples.
- **PM2-10k**: PM2-10k dataset contains 10,000 pairs of SMILE strings and binary promiscuous class labels.

Comparison Methods We compared our seq2seq fingerprint method with two state-of-the-art methods: the ECFP [56] (circular fingerprint) and the neural fingerprint method [52]. The circular fingerprint is a hand-crafted hash-based feature. The neural fingerprint is constructed on a supervised deep graph convolutional neural network. The circular fingerprint was generated through RDKit [1] and we use Multi-Layer Perceptron for the future predictive task as suggested in [52]. We obtained the neural fingerprint from `https://github.com/HIPS/neural-fingerprint` and we carefully followed the authors' instructions to apply our datasets.

Infrastructure and Software The seq2seq fingerprint method was implemented through tensorflow package [85], and the trained models used in our experiments were trained on a workstation with Intel i7 6700K @ 4.00 GHz CPU, 16 Gigabytes RAM and a Nvidia GTX 1080 GPU. We performed the hyper-parameter grid search and the training process of the classifiers on the TACC Lonestar 5 cluster [2]. In addition to the traditional MPI package for distributed grid-search, we used a more flexible master-worker task distribution package for Python called *dgsearch*. The code for training the seq2seq fingerprint will become publicly available after the acceptance of this paper.

4.4.2 Seq2seq Fingerprint Recovery Performance

Throughout the entire experiment sections, we discuss three variants of our seq2seq fingerprint, varying in feature vector lengths as 512, 768, and 1024. Each model was trained for 24 hours. These three models differ only in the number of GRU layers and yet with the same Latent Dimension (LD). We report the training details and the recovery power of each fingerprint model in Table 4.4. The recovery

[1] http://www.rdkit.org

[2] https://www.tacc.utexas.edu/systems/lonestar

performance is evaluated through the perplexity and Exact Match (EM) accuracy. The perplexity is calculated by the entropy of the probability distribution over the training set. The EM accuracy is the ratio between the exactly recovered SMILE strings and the total number of SMILEs in the test sets.

Table 4.4: The reconstruction performance with different number of GRU layers.

Model	Layer	LD	Perplexity	EM Accuracy
seq2seq-512	2	256	1.00897	94.24%
seq2seq-768	3	256	1.00949	92.92%
seq2seq-1024	4	256	1.01472	90.26%

Table 4.4 reveals a decreasing trend of recovery performance when we increase the layer number of stacked GRU cells. One might expect a deeper GRU network to have a better EM accuracy, which contrasts with the observation. The reason might be complex. First, the training of longer seq2seq fingerprint might take longer time to have better performance. Also, increasing the length of fingerprint actually expands the representation space of molecules, leaving more null spaces in the fingerprints. However, this observation does not indicate a longer fingerprint will decrease the performance in other tasks, as shown in the next subsection.

4.4.3 LogP Solubility and PM2 Promiscuous Classification

In this section, we report the classification performance of all three seq2seq models with fingerprint lengths 512, 768, and 1024, compared with the circular fingerprint [56] and neural fingerprint [52]. We use three ensemble classifiers for our seq2seq fingerprints: Adaboost [86], GradientBoost [87], and RandomForest [88]. We report the accuracy means and standard deviations of 5-fold classification cross validation on both LogP and PM2-10k data, in Table 5.1 and 4.3 respectively. All results

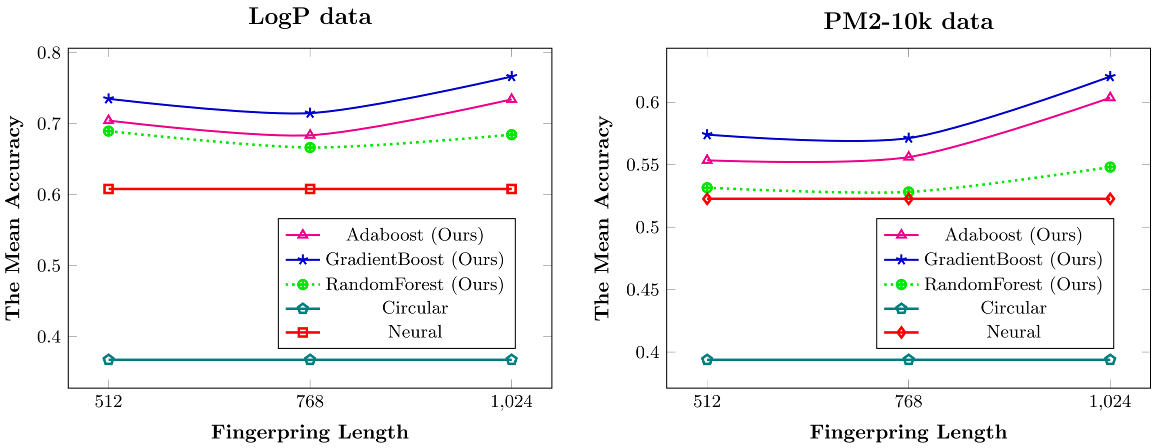


Figure 4.5: The mean accuracy of five methods. AdaBoost, GradientBoosting and RandomForest our methods. The circular and neural fingerprint are the state-of-the-art methods.

are the 100-run averages to reduce the randomness. We also show the impact of seq2seq fingerprint length on the accuracy in Figure 4.5.

From the Table 5.1 and 4.3, we observe, on both data sets, our methods significantly outperform the circular and neural fingerprints, regardless of classifiers and fingerprint lengths. The circular fingerprint is hashing-based and abandons a large portion of information and is not invertible to original molecule, while our fingerprint is completely invertible and encodes rigorous information. One might argue if the ensemble classifiers will improve the performance of circular fingerprint. According to [52] and our preliminary experimental observation, the results will be worse if we switch the MLP classifier to ensemble classifiers, e.g., GradientBoost, due to the limited length and information of circular fingerprint. The neural fingerprint is a supervised deep learning-based algorithm, and it could be highly limited by the amount of labeled training data. While our method transfers knowledge from a fairly large amount of unlabeled data, our method could outperform the neural fingerprint method in classification tasks. Overall, our seq2seq fingerprints encode rigorous information for molecules and could train on a huge amount of data to achieve task-insensitive performance.

In Figure 4.5, however, despite the lower recovery performance of seq2seq-1024 fingerprint, it does always provide the best classification performance, while, surprisingly, the seq2seq-768 seems to always have lower classification performance. The longer fingerprints might have more information for ensemble classification methods, but might also bring in noise. While the noise takes the major effects, the performance might decrease. But when the information is encoded enough, the performance will boost.

4.5 Conclusions

In this paper, we discuss a new unsupervised molecular representation system, called **seq2seq fingerprint**, based on the idea from the recent breakthrough on the English-to-French language translation, named sequence to sequence learning model. Our model translates the molecular SMILE string to the SMILE itself, while at the same time generates a fixed length fingerprint vector. The experiments on classification task demonstrate its superior performance. Also, the nature of our data-driven label-free model brings us even more benefits. 1) Our fingerprint system is completely unsupervised, meaning it will never be limited by the expensive label collection process. In fact, it could utilize each of every valid molecule, theoretically reaching the amount of infinite. 2) Contrast to the supervised learning models trained with very limited data samples, the seq2seq fingerprint is trained from a sufficiently large pool of samples, and therefore it is more robust to the specific task.

This seq2seq fingerprint is definitely not the end. It widely opens tons of new possibilities. Also due to the long training time, we might introduce efficient distributed training strategy [3, 85]. There are still many hyper-parameters in our training algorithms, in the future, we might want to pick an optimal method for hyper-parameter tuning [89]. Another quick future work would lie on how to embed some label information [90] to the fingerprint training to enhance its performance on the future machine learning task. Those type of semi-supervised learning could be a trend in the future drug discover tasks.

CHAPTER 5

SEQ3SEQ FINGERPRINT: IMPROVING DEEP SUPERVISED DRUG DISCOVERY WITH UNSUPERVISED TRAINING

Observing the recent progress in Deep Learning, the employment of AI is surging to accelerate drug discovery and cut R&D costs in the last few years. However, the success of deep learning is usually attributed to large-scale clean high-quality labeled data, which is generally unavailable in drug discovery practices.

In this chapter, we address this issue by proposing an end-to-end multi-task deep learning framework in a semi-supervised learning fashion. Given the enormous availability of unlabeled drug-like molecular data-sets, we reveal that significant improvement can be observed when employing large-scale unlabeled data and an auxiliary unsupervised self-recovery task. Compared with previous state-of-the-arts, the proposed method, named as **seq3seq fingerprint**, trains in a mixed data pool with both unlabeled and labeled data. Furthermore, an auxiliary unsupervised self-recovery task (loss) is coupled with specific inference tasks to regularize the supervised training, e.g., molecule solubility, promiscuousness, etc. Extensive experiments confirm the significant improvements over a variety of drug data-sets and demonstrate the effectiveness of the proposed techniques.

5.1 Introduction

In the past few years, the application of Artificial Intelligence (AI) technologies in drug discovery has become significant and increasingly popular. Observing the most recent rapid growth of a key technology in AI, namely **deep learning** (or **deep**

neural network), the whole industry and academia are looking towards AI to speed up the drug discovery, cut R&D cost and decrease the failure rate in potential drug screening trials [12].

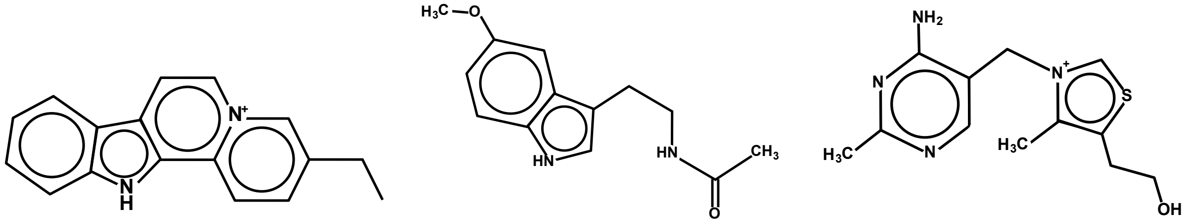
However, the previous success of deep learning in multiple applications, e.g., image understanding [13, 14], medical imaging [15, 3, 16], video understanding [17, 18], bioinformatics [10, 19, 20], and machine translation [21], etc., has implied a reliance on large-scale high-quality labeled data-sets. The training procedure of those deep-learning-based state-of-the-art models generally involve millions of labeled samples. In the meantime, however, for the drug discovery tasks, the scale of labeled data-set stays around only thousands of examples due to the insanely high cost of obtaining the clean labeled data through the biological experiments. The available amount of the labeled training data is absolutely insufficient to secure the success of the application of deep learning in the drug discovery. This huge gap between the requirement and availability of the labeled data in drug discovery has become a bottleneck of applying deep learning techniques into drug discovery.

Given the high cost of obtaining sufficient labeled data points, it seems impractical to increase the labeled data-set scale to a satisfactory level. To address this issue, we propose a semi-supervised deep learning modeling strategy. In simple terms, the proposed deep learning framework can learn from both labeled and unlabeled data, while the unlabeled data is almost infinitely available. For instance, the ZINC data-set [91] is publicly available and contains over 35 million unlabeled molecule data. With such scale of data being used, the deep learning model is expected to be trained with enough representation power to help the inference task.

In this paper, we propose a semi-supervised data-driven multi-task deep-learning-based drug discovery method, named as **seq3seq fingerprint**. The reasons behind this naming are two-fold: 1) this is the **next-generation seq2seq fingerprint** [10], whose major upgrade is that the original two-stage pipeline has been combined into an multi-task one-stage end-to-end pipeline to ensure much more decent inference performance; 2) the seq**3**seq fingerprint framework contains **three** ends with one input and two outputs while the seq**2**seq fingerprint contains **two** ends with one input and one output.

To briefly introduce the proposed seq3seq fingerprint framework, the seq3seq fingerprint network can be considered as a pipeline with one input and two outputs. The designed neural network can take the molecule inputs for training, **with or without labels**. The input is the raw sequence representation of a molecule, namely SMILE representation. Examples are referred in Figure 5.1. The two outputs will correspond to the two tasks inside this network. The first one is the **self-recovery**. The network is expected to be able to generate a vector representation which is able to be recovered back to original raw sequence representation. The second task is the **inference** whenever the label is available. For instance, it can be a task to predict the acidity, alkalinity or solubility of a single molecule. The two tasks are trained within the same network in an end-to-end fashion. As a result, in a specific inference task, the vector representation will be able to provide both good recovery performance and inference performance. Also, the network can be trained inside a mixture data pool with both labeled and unlabeled data, which is sufficient enough to ensure the fine training of the neural network.

The benefits of the seq3seq fingerprint are three folds: 1) the training phase of seq3seq fingerprint takes both labeled and unlabeled data into consideration, which is able to provide both strong vector representation and good inference performance. 2) it is data-driven, eliminating the reliance on expert's subjective knowledge. 3) since the unlabeled data is almost unlimited in practice, it will significantly complement the sole training with labeled data, ensuring a final good inference performance.



(a) Flavopereirin (b) Melatonin (c) Thiamine

Figure 5.1: The examples of SMILE representations.

The technical contributions of this paper are summarized as: 1) the seq3seq fingerprint method is obviously the first attempt to utilize both labeled data and unlabeled data for sequence-based end-to-end deep learning in drug discovery. 2) We reveal that the following unsupervised training techniques gain largely on the supervised deep drug discovery task performance:

- **mixed unlabeled and labeled data training** training on a mixture of both unlabeled and labeled data can significantly improve the final inference results.
- **unsupervised task** it is beneficial for the inference training to involve a self-recovery unsupervised task.

3) Extensive experiments demonstrate the superior performance on different tasks over both supervised and unsupervised state-of-the-art fingerprint methods.

The rest of the paper is organized as follows. We summarize several related work in drug discovery, in Section 5.2. In Section 5.3, we describe our entire pipeline in details. We show our experiment results in Section 5.4, demonstrating the superior performance of our method. We conclude and discuss the future direction of our paper in Section 5.5.

5.2 Related Work

In this section, we briefly introduce several related works. First, we present the raw representation of molecules, namely SMILE representation, i.e., the persistence form of the molecular data in the cold data storage. Second, we list a few state-of-the-art fingerprint methods, including the ones using human-designed and hash-based features.. Finally, we briefly describe some most recent deep learning based methods, e.g., neural fingerprint [82], seq2seq fingerprint [10].

5.2.1 SMILE Representations of Molecules

5.2.1.1 Vanilla SMILE Representation System

Initially, the molecules are stored in the form of a sequence representation, namely the Simplified Molecular-Input Line-Entry system (SMILE) [75], which is a line notation for describing the structure of chemical species using text strings. The SMILE system represents the chemical structures in a graph-based definition, where the atoms, bonds and rings are encoded in a graph and represented in text sequences. Simple examples of SMILE representations are 1) dinitrogen with structure $N \equiv N$ (N#N), 2) methyl isocyanate with structure $CH_3 - N = C = O$ (CN=C=O), where corresponding SMILE representations are included in the brackets. Simply speaking, the letters, e.g., C, N, generally represent the atoms, while some symbols like $-, =, \#$ represent the bonds. We show some more complicated examples in Figure 5.1.

5.2.1.2 Canonical SMILE: Bijective Mapping Between SMILEs and Molecules

SMILE system is not perfect in that the vanilla SMILE system is not a bijective mapping between SMILE sequence and a molecule. For example, a molecule can have multiple corresponding SMILE representations, e.g., CCO, OCC and $C(O)C$. To address this issue and providing a one-to-one mapping between SMILEs and molecules, multiple canonicalization algorithms are invented to ensure the representation uniqueness of each molecular structure [92]. In this paper, all of our SMILEs are canonical SMILEs to ensure the bijectiveness of the mapping.

5.2.2 Fingerprint Methods

Traditionally, there is a major class of molecular representation system called **fingerprint**. A fingerprint is basically a vector of a corresponding molecule as its continuous representation. Hence fingerprints can be thereafter fed into a machine learning system as an initial vector representation. A large number of previous studies are inventing new fingerprint systems which can benefit future predictive tasks.

5.2.2.1 Hash-based Fingerprints

Many hash-based methods has been developed to generate unique molecular feature representation [53, 54, 55]. One important class is called **circular fingerprints**. Circular fingerprints generate each layer's features by applying a fixed hash function to the concatenated features of the neighborhood in the previous layer. One of the most famous ones is Extended-Connectivity FingerPrint (ECFP) [56]. However, due to the non-invertible nature of the hash function, the hash-bashed fingerprint methods usually do not encode enough information and hence result in lower performance in further predictive tasks.

5.2.2.2 Biologist-guided Local-Feature Fingerprints

Another mainstream of traditional fingerprint methods is designed based on the biological experiments and the expertise knowledge and experience, e.g., [57, 58]. Biologists look for several important task-related sub-structures (fragments), e.g.,

$CC(OH)CC$ for pro-solubility prediction, and count those sub-structures as local features to produce fingerprints. This kind of fingerprint methods usually work well for specific tasks, but poorly generalize for other tasks.

5.2.3 Deep-learning-based Models

The growth of deep learning has provided the great flexibility and performance to create the molecular fingerprint from data samples, without explicit human guide, [52, 50, 51, 76, 77, 10]. In this subsection, we discuss two major classes, namely supervised and unsupervised learning models.

5.2.3.1 Supervised Models

Many of deep learning-based fingerprint methods are still trained in a supervised-learning fashion [76, 93], which is using only labeled molecular data samples as inputs and adjusting model weights according to their labels [94]. However, as mentioned earlier, the performance of the deep supervised learning models are generally limited by the availability of the labeled data. The state-of-the-art work is the neural fingerprint [52]. The neural fingerprint mimics the process of generating circular fingerprint but instead the hash function is replaced by a non-linear activated densely connected layer. This method is based on the deep graph convolutional neural network [72, 95, 96, 94]. There are also few attempts that address the insufficient label issue by using few-shot learning strategies, e.g., [97]. To secure a satisfactory performance and acquire enough labeled data, biologists need to perform a sufficiently large number of tests on chemical molecules, which is prohibitively expensive.

5.2.3.2 Unsupervised Models

Recently, few unsupervised fingerprint methods, e.g., seq2seq fingerprint [10], are proposed to alleviate the issue brought by the insufficient labeled data. These models generally train deep neural networks to provide strong vector representations using a big pool of unlabeled data. The vector representation model is thereafter used for supervised training with other models, e.g., Adaboost [86], GradientBoost [87], and RandomForest [88], etc. Since the deep models are trained with a sufficiently large data-set, the representation is expected to contain enough information to provide good inference performance. However, this type of methods are not trained end-to-end, meaning that the representation only adjusts to the recovery task of the original raw representation. It is robust to the specific labeled task, but might not provide optimal inference performance for each task.

5.2.4 Relationship with Natural Language Processing Models

One might spot that the modeling with sequence-based drug discovery shares many similarities with the Natural Language Processing (NLP) modeling. For example, they both use famous Recurrent Neural Network (RNN) techniques as state-of-the-art methods in deep learning-related methods. However, the problem itself in drug discovery might be quite different from that in NLP area. First, the vocabulary in drug discovery tends to be much smaller but much less related. For example, token C (carbon) is basically no same as token O (oxygen) while, in NLP, the word ”man” and ”woman” can be embedded to a space where they can be close to each other using techniques like word2vec [98]. Also, in general, say in English, the length of a sequence (or sentence) in NLP is generally shorter than that in drug discovery. A drug-like SMILE sequence in drug discovery can generally size around 100 tokens, while English sentences usually contain 15-20 words each [1]. This brought significant challenges to some models, especially Long Short-term Memory (LSTM) which uses a forgot gate to constantly ”forget” older memory, which is absolutely not ideal for long sequences.

We have been most recently aware of a similar work in NLP area [99]. This work comes out mostly in parallel with our work. This paper tackles multiple tasks in NLP area, e.g., textual entailment, question answering, semantic assessment, and document classification. They reveal that large gains can be realized by generative pre-training on a large pool of unlabeled text corpora, which, to some extent, confirms our solid work of the mixed training on both unlabeled and labeled data. Dating back earlier, there is also some work in semi-supervised learning for sequence learning [100], or adversarial training on sequence [101].

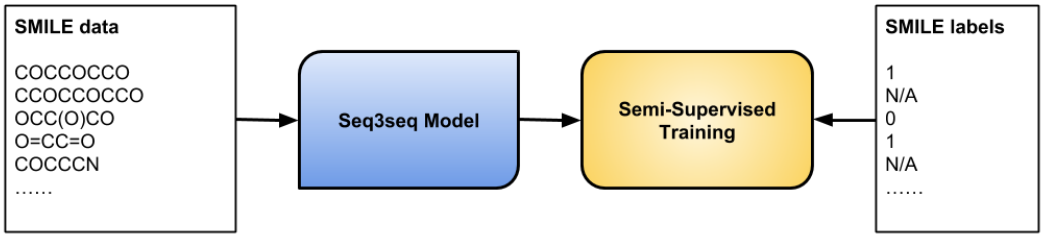


Figure 5.2: This figures shows how semi-supervised training is used for our proposed model. We mix the unlabeled data and labeled data together to train our proposed model. The SMILEs with label 0/1 come from labeled dataset and the SMILEs without labels (N/A in the figure) come from unlabeled dataset.

[1] https://strainindex.wordpress.com/2008/07/28/the-average-sentence-length/

5.3 Methodology

In this section, we describe the details of our semi-supervised seq3seq fingerprint model. First, an overview of the proposed seq3seq fingerprint model is given. The proposed semi-supervised model is trained in an end-to-end fashion by completing two tasks, a self-recovery task for molecule (without any label) and an inference task (with specific classification/regression label). Second, we will detail the perceiver network, the self-recovery and the drug discovery task-specific loss, respectively. Finally, we provide a discussion of different views of the proposed framework, e.g., from a multi-task scaffolding view from frame-semantic parsing [102] in natural language processing area.

5.3.1 Overview

Different from traditional models [82, 10], the proposed seq3seq fingerprint model works in a semi-supervised fashion. It means that our training data comes from two sources, the labeled data, for classification/regression, as well as the unlabeled data. The labeled data contains the SMILE strings for molecule data and their labels, such as acidity or other molecular activities. The unlabeled data contains just molecular SMILE strings and the unlabeled data is almost infinitely available. The proposed seq3seq fingerprint model takes the mixture of the labeled data and unlabeled data together as training inputs to the network. The work flow is depicted in Figure 5.2. The semi-supervised training is done by two tasks: the self-recovery task and the inference task. The whole pipeline is illustrated in Figure 5.3.

5.3.2 Perceiver Networks

In this subsection, we introduce more details about the neural network designed to encode a SMILE sequence to a feature vector, i.e., perceiver network. As a sequence-encoding network, we naturally use Recurrent Neural Network (RNN)-based structure for the perceiver network. Specifically, in our work, we use both LSTM and GRU units in our experiments. We then detail each of them.

5.3.2.1 LSTM Units

The Long Short-Term Memory (LSTM) [103] is the most widely used recurrent neural network units. The LSTM Units have three gates: input gate, forgot gate, and output gate. A LSTM network computes a sequence of network outputs $(s_1, \ldots, s_T)$ from the input sequences $(x_1, \ldots, x_T)$ by iterating

$$\begin{aligned}
f_t &= \sigma_g(W_f X_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma_g(W_i x_t + U_i s_{t-1} + b_i) \\
o_t &= \sigma_g(W_o x_t + U_o s_{t-1} + b_o) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c + U_c h_{t-1} + b_c) \\
h_t &= o_t \circ \sigma_h(c_t). \qquad (5.1)
\end{aligned}$$

The LSTM cell has a "forgot" gate f_t which is to block some of the previous state to pass through the entire sequence. i_t and o_t are the input and output gates for the LSTM cell at time step t. c_t and h_t are the LSTM **cell** state and **hidden** state. σ represents activation function. σ_g is usually set to sigmoid function, while σ_c and σ_h are usually the hyperbolic tangent functions.

5.3.2.2 GRU Units

The Gated Recurrent Unit (GRU) [104] is used in our experiment. GRU is famous for its LSTM-comparable performance but faster training process. A

GRU network computes a sequence of outputs $(s_1, \ldots, s_T)$ from the input sequences $(x_1, \ldots, x_T)$ by iterating

$$\begin{aligned} z_t &= \sigma_g(W_z x_t + U_z s_{t-1} + b_z) \\ r_t &= \sigma_r(W_r x_t + U_r s_{t-1} + b_r) \\ h_t &= \tanh(U_h x_t + W_h(s_{t-1} \circ r_t)) \\ s_t &= (1 - z_t) \circ h_{t-1} + z_t \circ s_{t-1}. \end{aligned} \tag{5.2}$$

A GRU cell has two gates: the update gate z and the reset gate r. Each gate has the trainable parameter W, U, b. The activation function σ for each gate is usually the sigmoid function. GRU also has the *"hidden memory"* h, which holds another set of trainable parameters U, W. In contrast with LSTM which has three gates, it has similar performance but faster training speed [83].

5.3.3 The Duo Tasks in Seq3seq Fingerprint Model

In this subsection, we introduce the multi-task training of the proposed framework. The framework incorporates two tasks for training: self-recovery and molecular inference tasks. Though the major task in drug discovery might be molecular inference, the self-recovery task can help the training as well because it builds up a stronger and richer vector representation of the SMILE sequence. Finally, we introduce the overall training procedure for the multi-task model framework.

5.3.3.1 The Self-recovery Task

The self-recovery task is to learn a strong, rich vector representation (usually noted as **fingerprint** in the drug discovery literature) for each input molecular SMILE string by allowing the vector representation to be recovered to the original representation, i.e., SMILE sequence. It is an **unsupervised learning** task since no task-specific label information is used in training. As shown in Figure 5.3, this task will use the perceiver network and adds an interpreter network to recover the original SMILE sequence from the fingerprint vector. This structure is motivated by the seq2seq model [10, 81]. The original seq2seq model is used in machine translation [81]. It is to learn a vector representation from a sentence in a given language, e.g., English, then translate the learned representation into another language such as French. Seq2seq fingerprint [10] combines the idea from seq2seq learning and the idea of auto-encoder to learn the vector representation for molecule.

The neural network used to recover the original SMILE sequence, namely **interpreter network**, shares similar fundamental parts with the perceiver network, i.e., Recurrent Neural Network (RNN). In this paper, we limit our scope to use same type of RNN for both perceiver and interpreter networks. However, it is worth to mention they can be selected to different types of RNNs, e.g., LSTM for perceiver network and GRU for interpreter network as long as their internal state sizes match. The proposed framework thus allows a great extent of flexibility for model design.

For self-recovery task, we use the **sparse cross entropy** loss. The token vocabulary $\{v_1, v_2, \ldots, v_N\}$ of SMILE sequence is unique and limited. Set $z_t \in \mathbb{R}^N$ as the output token distribution from the RNN cell outputs, and $l_t \in \mathbb{R}^N$ as the one-hot vector of the given original SMILE sequence token at time step t. Thus the unsupervised loss $\mathcal{L}_{unsup}$ is given by:

$$\mathcal{L}_{unsup} = \sum_{t=1}^{T} l_t^T \log(z_t). \tag{5.3}$$

The crux of this loss is a sum of sparse cross entropy loss of each SMILE sequence token.

5.3.3.2 The Molecular Inference Task

The inference task in the proposed seq3seq fingerprint model is to predict the activity of molecules. In the proposed model, the inference task includes the perceiver network and the inference network. The perceiver network is shared in both self-recovery and inference tasks. It is trained by both labeled and unlabeled data in an end-to-end fashion. The inference network maps the seq3seq fingerprint to a final inference result on a certain prediction task. The structure of the inference network can be any trainable network which maps the vector into a inference value. It allows huge flexibility for the choice of the inference network. For instance, it could be a Convolutional Neural Network (CNN), a Multi-Layer Perceptron (MLP) or even a single fully-connected layer.

Depending on whether the inference task is classification or regression, the loss for the inference task $\mathcal{L}_{sup}$ could be either classification loss (usually a cross entropy loss) or regression loss (usually a ℓ_1 smooth/ℓ_2 distance loss). Since computing the $\mathcal{L}_{sup}$ needs labels, the inference task is only trained on labeled data.

5.3.3.3 End-to-end Semi-supervised Learning

As shown in Figure 5.3, the semi-supervised loss $\mathcal{L}_{semi}$ combines the unsupervised loss $\mathcal{L}_{unsup}$ and the supervised loss $\mathcal{L}_{sup}$ together as

$$\mathcal{L}_{semi} = \begin{cases} \mathcal{L}_{unsup} + \lambda\mathcal{L}_{sup}, & \text{if label is present,} \\ \mathcal{L}_{unsup}, & \text{if label is not present} \end{cases} . \tag{5.4}$$

where λ is a hyper-parameter of the proposed model to balance the two tasks. The proposed model is trained with both supervised data and unsupervised data together. When the data is unlabeled, the supervised loss $\mathcal{L}_{sup}$ will be zero. Thus, in this case, only the part of the model in self-recovery task will be trained. While the data is labeled, both the part of the model in self-recovery and inference will be trained. The end-to-end training avoids the model from pre-trained model or separated classifier [10]. As a result, the proposed end-to-end model is expected to provide an optimal inference performance for specific task than that in a multi-stage model from [10].

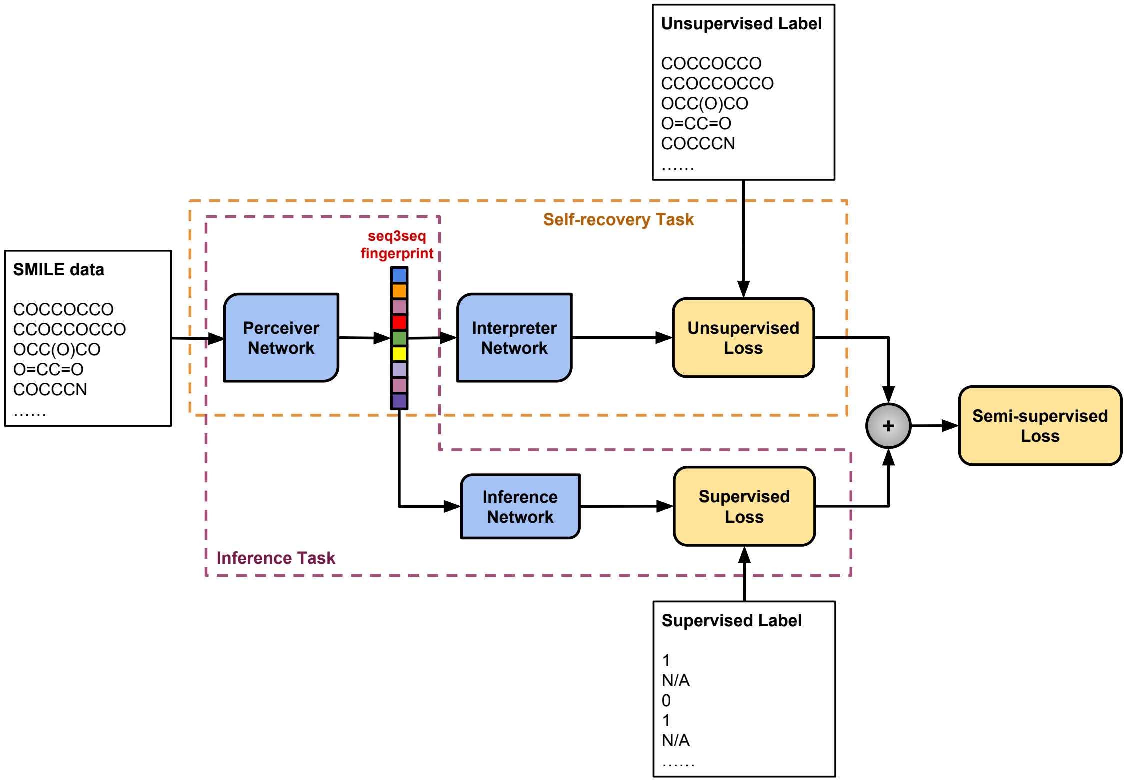


Figure 5.3: This figure shows the proposed seq3seq fingerprint model. The proposed model is trained through two tasks: a self-recovery task and an inference task. The self-recovery task contains a perceiver network and an interpreter network; the inference task shares the perceiver with self-recover task and has an inference network. The semi-supervised loss is the sum of supervised loss and unsupervised loss.

5.3.4 Discussion

5.3.4.1 A Multi-task Scaffolding View of Seq3seq Fingerprint

In [10], the authors viewed seq2seq fingerprint as a machine translation problem in the Natural Language Processing (NLP) area, with both source and target language set to be the SMILE representation. Interestingly, the proposed seq3seq fingerprint model can be viewed, to some extent, as **a multi-task scaffolding framework** [102] in the NLP area as well. In [102], the authors focus on solving the frame-semantic parsing problem, which is basically finding the *action* (frame) with its associated objects from a sentence. For example, in sentence "Alice loves Bob.", the frame is "loves" with its associated objects being "Alice" and "Bob". However, a single sequence-to-frame network model generally performs poorly in this task. In [102], they proposed to use a multi-task framework to refine the predictions. Besides the frame parsing task, they also introduce the syntactic parsing task. The second task is basically predicting the word categories, e.g., nouns, adverbs, adjectives, etc. For the previous "Alice loves Bob." sentence, the result will be that "Alice" being noun, "loves" being verb and "Bob" being another noun. In [102], it is demonstrated that the second task significantly helps the success of the main (frame parsing) task. To sum up, the multi-task scaffolding frame parsing framework utilizes a second *syntactic parsing* task to reinforce the main task which is the *frame parsing.* Our seq3seq fingerprint can be viewed in a very similar fashion: the **self-recovery task** serves as the auxiliary task to augment the main **prediction task**. This modification is also further demonstrated superior in our experiments described in Section 5.4.

5.4 Experiments

In this section, we first detail the experimental setup, e.g., the data set description, hardware and software settings, etc. Then we report the benchmark performance of the seq3seq fingerprint methods among state-of-the-art methods. Furthermore, to show the flexibility of our methods and complete our experiments, we offer ablation studies for the sensitivity of the hyper-parameters of our seq3seq fingerprint models, e.g., the multi-task balance weight λ, the Recurrent Neural Network (RNN) layer hidden size and layer number, RNN cell type, etc.

5.4.1 Experiment Setup

Table 5.1: The comparison of classification accuracy on the LogP data. We report the average classification accuracy (Mean) and the corresponding Standard Deviation (StDev) of 5-fold cross-validation result.

	Circular [56]	Neural [52]	seq2seq [10]	seq3seq (Ours)
Mean	36.74%	60.80%	76.64%	**89.72%**
StDev	0.74%	1.35%	0.43%	0.41%

Table 5.2: The comparison of classification accuracy on the PM2-10k data. We report the average classification accuracy (Mean) and the corresponding Standard Deviation (StDev) of 5-fold cross-validation result.

	Circular [56]	Neural [52]	seq2seq [10]	seq3seq (Ours)
Mean	39.38%	52.27%	62.06%	**68.45%**
StDev	1.14%	1.12%	1.98%	0.80%

Datasets As we mentioned in the introduction, the seq3seq fingerprint can be trained from a mixture of both unlabeled and labeled data. In practices, we usually use an unlabeled data set of a much larger size than that of a labeled dataset.

Unlabeled Dataset For (large) unlabeled dataset, we use ZINC drug-like datasets [91]. ZINC is a free database of commercially-available compounds for virtual screening. The drug-like dataset from ZINC contains 18,691,354 molecular SMILE representations.

Labeled Dataset Two additional datasets, LogP and PM2-10k, were used for semi-supervised training and test. They are obtained from National Center for Advancing Translational Sciences (NCATS) at National Institutes of Health (NIH). Each of them contains around 10,000 molecular SMILE representations with multiple scores, each score quantifies some chemical property. Classification was conducted on LogP and PM2-10k.

- **LogP**: Totally 10,850 samples were used from LogP, Each sample contains a pair of a SMILE string and a water-octanol partition coefficient (LogP) value. A threshold of 1.88 is used to label the data. For those samples with LogP value smaller than 1.88 were classified as negative samples, the rest were labeled as positive samples.
- **PM2-10k**: PM2-10k dataset contains 10,000 samples of SMILE strings and binary promiscuous class labels. Similarly, a threshold of 0.024896 was used to classify each SMILE. Samples with value larger than the threshold were considered as positive 1; otherwise, labeled as 0.

We mix the ZINC drug-like dataset with the labeled dataset and train the recovery and inference task simultaneously on the mixed dataset.

Neural Network Structures As we mentioned earlier, the proposed seq3seq fingerprint framework is super flexible in the choice of the network structure. Theoretically, both perceiver and interpreter network can use any stacked Recurrent Neural Network (RNN) with different layers and layer hidden sizes. Also the RNN cell can be formed in different types, e.g., LSTM, GRU, etc. Due to the page limit of this paper, we hereby assume the perceiver and interpreter network always use the same type of RNN cells with the same number of layers and hidden sizes. In this section, we discuss different types of RNN cells, e.g., GRU [104], LSTM [103], etc. Also, we limit the discussion of the inference network to a single densely connected layer with the output number equaling the number of the classification class number. For simplicity, we use $GRU - L - H$ to represent the network structure, where GRU is the RNN cell type (it can be $LSTM$ as well), $L \in \mathbb{N}^+$ is the stacked RNN layer number and $H \in \mathbb{N}^+$ is the RNN cell hidden size. For instance, $GRU-2-256$ represents a seq3seq model where both perceiver and interpreter network use 2-layer GRU cell with 256 hidden units.

Learning Hyper-parameters For optimization, we use the Stochastic Gradient Descent (SGD) with a heuristic learning rate decaying schedule. The initial learning rate is 0.5 for any training models. The learning rate will be decayed by a factor of 0.99 if the test loss does not decrease after 600 training steps. The training will automatically halt if the learning rate is smaller than $1e-7$. Under the above hyper-parameter sets, the training of each model in the semi-supervised setting can generally finish within a few hours.

Evaluation Metrics Given that we have two tasks of our semi-supervised learning framework, i.e., recovery and inference task, we report two evaluation metrics for each model we trained. For recovery task, we use an Exact Match Accuracy (EMA) for evaluation. This metric measure the portion of the exactly recovered sequence within the entire set of sequences. Furthermore, we report the classification accuracy (hereafter SSLA for Semi-Supervised Learning Accuracy) for our classification task.

Comparison Methods We compare our semi-supervised method with the unsupervised seq2seq fingerprint method [10] as well as several other state-of-the-art methods: the ECFP [56] (circular fingerprint) and the neural fingerprint method [52]. We download the official implementation of the seq2seq fingerprint [2] and carefully follow the experimental setting of the authors. The circular fingerprint is a hand-crafted hash-based feature that was generated through RDKit [3]. The neural fingerprint implementation is obtained from `https://github.com/HIPS/neural-fingerprint`, which we slightly modify to adapt our dataset file format.

Infrastructure and Software The seq3seq fingerprint method was implemented through Tensorflow package [85], and our semi-supervised model was trained in a self-hosted 16-GPU cluster platform with Intel i7 6700K @ 4.00 GHz CPU, 64 Gigabytes RAM and four Nvidia GTX 1080Ti GPUs on each workstation. The code will be released upon the acceptance of this paper.

5.4.2 Comparison with State-of-the-art Methods

In Table 5.1 and 5.2, we report the 5-fold cross validation average classification accuracy on LogP and PM2-10k datasets. The proposed methods are compared with ECFP (circular) fingerprint [54], neural fingerprint [82] and seq2seq fingerprint [10]. For seq2seq fingerprint, according to their paper, the seq2seq fingerprint with length 1024 + Gradient Boosting always provides best performance, so we only report those results on our paper.

It is shown that on both datasets, the seq3seq fingerprint always provides best inference performance. On LogP dataset, our seq3seq model performs significantly superior than the other state-of-the-art methods, up to 13% in terms of classification accuracy (SSLA in the tables). Compared with circular fingerprint, the seq3seq fingerprint is data-driven and contains enough information to be recovered. The performance of neural fingerprint is generally limited by the availability of the labeled

[2] https://github.com/XericZephyr/seq2seq-fingerprint

[3] http://www.rdkit.org

data. Seq2seq fingerprint is the closest work in terms of accuracy for now since it can be also trained on the huge pool of unlabeled data, extracting a good representation and train/infer with a sophisticated classification model. However, seq2seq fingerprint is, unfortunately, not an end-to-end framework, which means the recovery and inference training of seq2seq fingerprint are separate. The unsupervised recovery training can bring in considerable amount of noise in the representation which limits further improvements of the inference performance. The seq3seq fingerprint, which uses the inference task to correct the recovery task during training, can constantly provide the best performance among all of the comparison methods.

Table 5.3: The performance variations with λ and GRU model parameters for LogP data. Layer: the stacked layer number of RNN cells. LD: Latent Dimension (hidden size) of RNN cells. EMA: Exact Match Accuracy for self-recovery task. SSLA: classification accuracy for inference task.

Layer	LD	λ	EMA	SSLA
2	128	1	86.31%	89.46%
		0.1	91.80%	89.62%
		0.01	90.23%	81.05%
		0.001	91.42%	64.95%
2	256	1	93.59%	90.18%
		0.1	94.52%	89.35%
		0.01	95.77%	84.65%
		0.001	95.48%	69.16%

5.4.3 Sensitivity Analysis of Multi-task Weight Balance Parameters

In multi-task machine learning practice, the weight balancing hyper-parameters among different tasks (in our case, λ in the multi-task loss function (5.4)) are sometimes critical and sensitive to data. This might not be an intriguing feature in practices. However, our method is quite robust and tolerant with λ variations. In this

Table 5.4: The performance variations with λ and GRU model parameters for PM2-10k data. Layer: the stacked layer number of RNN cells. LD: Latent Dimension (hidden size) of RNN cells. EMA: Exact Match Accuracy for self-recovery task. SSLA: classification accuracy for inference task.

Layer	LD	λ	EMA	SSLA
2	256	1	87.48%	65.28%
		0.1	89.84%	64.85%
		0.01	91.73%	62.37%
		0.001	91.31%	50.66%
3	256	1	82.40%	64.90%
		0.1	87.61%	67.92%
		0.01	89.33%	68.24%
		0.001	90.25%	50.07%

Table 5.5: The comparison of 5-fold cross validation classification accuracy among different seq3seq GRU models on the LogP data. Both average (Mean) and Standard Deviation (StDev) are reported for the 5-fold splits. FP Length: FingerPrint Length. SSLA: classification accuracy for inference task. EMA: Exact Match Accuracy for self-recovery task.

	GRU-2-128	GRU-3-128	GRU-4-128	GRU-5-128	GRU-2-256	GRU-3-256	GRU-4-256	GRU-5-256
FP Length	256	384	512	640	512	768	1024	1280
SSLA Mean	89.62%	89.12%	89.05%	**89.72**%	89.48%	89.64%	88.90%	88.11%
SSLA StDev	0.62%	0.22%	0.10%	0.41%	0.44%	0.42%	0.31%	0.40%
EMA Mean	91.39%	85.75%	77.13%	68.64%	96.13%	94.24%	87.99%	83.86%
EMA StDev	0.46%	0.53%	0.56%	0.80%	0.21%	0.31%	0.45%	0.41%

Table 5.6: The comparison of 5-fold cross validation classification accuracy among different seq3seq GRU models on the PM2-10k data. Both average (Mean) and Standard Deviation (StDev) are reported for the 5-fold splits. FP Length: FingerPrint Length. SSLA: classification accuracy for inference task. EMA: Exact Match Accuracy for self-recovery task.

	GRU-2-128	GRU-3-128	GRU-4-128	GRU-5-128	GRU-2-256	GRU-3-256	GRU-4-256	GRU-5-256
FP Length	256	384	512	640	512	768	1024	1280
SSLA Mean	65.65%	67.11%	65.80%	67.23%	66.74%	68.08%	**68.45**%	67.09%
SSLA StDev	0.19%	0.85%	0.61%	0.52%	0.57%	0.35%	0.80%	0.67%
EMA Mean	83.84%	81.24%	78.60%	74.38%	92.49%	91.72%	87.36%	82.64%
EMA StDev	0.45%	0.67%	0.88%	0.88%	0.37%	0.25%	0.29%	0.76%

subsection, we report our sensitivity studies of λ. We choose different scale of λ to see how the final model performance responds to the variance of λ , showing the robustness of our method with regard to different weight balancing hyper-parameters.

In Table 5.3, 5.4 as well as Figure 5.5, we vary λ in the logarithm scale with a base of 10. We tried $10^0, 10^{-1}, 10^{-2}, 10^{-3}$. On both datasets, it looks that within a quite wide range of λ, i.e., $10^{-2} - 10^0$, the performance is quite robust to the change of λ. The reason behind this robustness might be the huge unlabeled data pool used in the training process. Given the model has been trained with a sufficiently large (up to dozens of millions) molecular data pool, the resulting model will automatically adjust to a small task weight perturbation.

5.4.4 The Ablation Study of Neural Network Structures

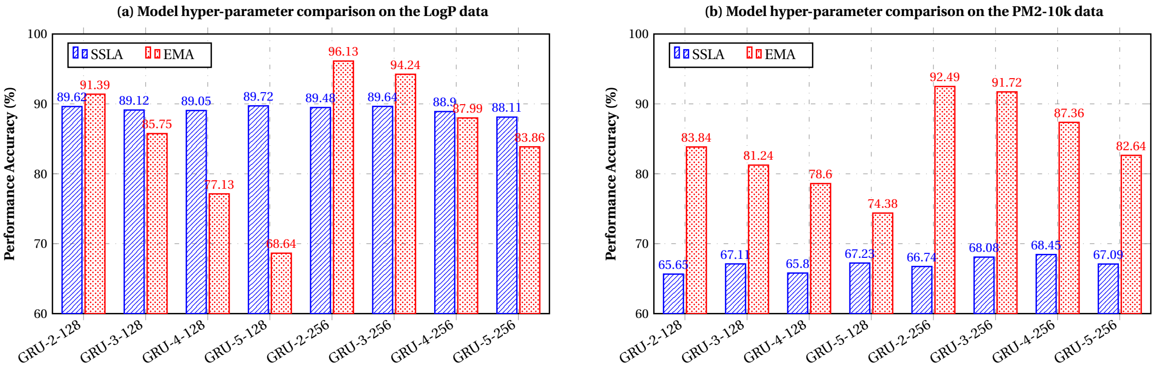


Figure 5.4: Impacts of the network structures on different metrics on both LogP and PM2-10k dataset. 1) The robustness of inference performance (SSLA, blue bars) is revealed. 2) The positive and negative correlations with regard to the self-recovery performance (EMA, red bars) are observed for RNN network depths and widths, respectively.

In this section, we provide a comprehensive study of the impacts of different layers and layer hidden sizes of our seq3seq fingerprint models. We report the 5-

fold cross validation Exact Match Accuracy (EMA) and the classification accuracy (SSLA) in Table 5.5 and 5.6 for each of the two datasets, respectively. Figure 5.4 (a) and (b) also illustrates the trends when varying the layer numbers and layer hidden sizes.

Inference Task It is super exciting to reveal the **robustness of classification accuracy to the change of network structures** on both datasets. In Figure 5.4, the classification accuracy (blue bars) almost stays at the same height when varying the layer numbers and layer hidden sizes. This implies the importance of the representation learning inside the seq3seq fingerprint. This further support the positive effects of the large-scale (up to dozens of millions) unlabeled data utilization.

When the inference is super robust to the network changes, for self-recovery task (in terms of EMA), we observe a decreasing trend when increasing the layer depth (numbers). Meanwhile, the increasing number of hidden units inside each layer generally yields better EMA. This suggests that the improvement of self-recovery task has higher reliance on the layer hidden sizes. Deeper network might not always be an elixir for a simple auxiliary task like self-recovery. This observation might help future network design. To simultaneously ensure high inference performance and reduce training time (deeper network generally takes longer to train.), it might be a good idea to use reasonably deep and wide RNN networks.

5.4.5 Influences of Different RNN Cells

In this section, we study the effects of varying different types of Recurrent Neural Network (RNN) cells. In Table 5.7 and 5.8, we show the results for GRU and LSTM on LogP and PM2-10k datasets. For the ease of presentation, we only show the results for 3-layer stacked RNN cells with hidden size set to 256. For the supervised inference task, it appears that GRU always performs better than LSTM,

Table 5.7: Comparison between different types of RNN cells with or without self-Recovery loss function on LogP data-set. SSLA: classification accuracy for inference task. EMA: Exact Match Accuracy for self-recovery task.

Network	SSLA	EMA
GRU w/o Rec Loss	89.47%	00.00%
GRU w/ Rec Loss	**91.41%**	56.65%
LSTM w/o Rec Loss	90.22%	00.00%
LSTM w/ Rec Loss	90.87%	**57.75%**

Table 5.8: Comparison between different types of RNN cells with or without self-Recovery loss function on PM2-10k data-set. SSLA: classification accuracy for inference task. EMA: Exact Match Accuracy for self-recovery task.

Network	SSLA	EMA
GRU w/o Rec Loss	54.87%	00.00%
GRU w/ Rec Loss	**63.91%**	49.38%
LSTM w/o Rec Loss	52.42%	00.00%
LSTM w/ Rec Loss	52.94%	**58.15%**

while for the unsupervised self-recovery task, LSTM is usually the winner. From this observation, we might conclude that LSTM generally performs better to memorize the original sequence while the GRU can learn the specific task faster and better.

We also switch the self-recovery task on and off and see the effects. It is observed that, without the recovery loss, the self-recovery is totally messy as the Exactly Match Accuracy (EMA) is zero. Also, when the self-recovery loss is appended, the supervised inference performance, in terms of SSLA, also always boosts (up to **9.04%** as shown in Table 5.8). It also demonstrates the promising results when applying self-recovery task to augment the supervised task.

5.4.6 Ablation Study of Large-scale Pre-training and Self-recovery Loss

In this section, we introduce the ablation study of two major components of the proposed framework, i.e., the unsupervised pre-training on a (relatively) huge unlabeled dataset and the self-recovery unsupervised auxiliary task. For simplicity, we fix model structure to a 3-layer stacked GRU units, each with hidden size 256. We compare the models with/without the unsupervised pre-training (pre/nopre) on a large unlabeled dataset and the models with/without the self-recovery task/loss (rec/norec). We show the results of the ablation study on both LogP and PM2-10k datasets in Table 5.9 and Table 5.10.

On both datasets, we can observe the best performance occurs when we utilize both techniques that we propose in this paper. Especially on PM2-10k data-set, we observe a huge improvement (up to 14.72% in terms of accuracy) when using both proposed techniques. For supervised (binary classification) task, we observe the utilization of a pre-training model on a big unlabeled dataset always helps the supervised task performance. This suggests proper use of a unsupervised pre-training model can even help unrelated supervised task. The unsupervised pre-training can provide a stronger and richer representation when training offline on a huge unlabeled dataset. It can help the further supervised predictive tasks.

Furthermore, we observe the self-recovery task, which can be trained in an unsupervised manner, can also augment the supervised predictive task. This indicates that representation learning on-the-fly can also benefit the inference task training. One might spot the models with self-recovery task and those without self-recovery tasks when using pre-training technique generally have very similar inference performance. But they diff a lot in self-recovery task. This implies the self-recovery task can make the inference task training escape the original local minima and have a better comprehensive performance.

Table 5.9: Ablative comparison of unlabeled data pre-training and self-recovery tasks on LogP data-set. ”rec” indicates the self-recovery task, ”pre” indicates the pre-training technique. Prepending ”no” means the opposite, e.g., ”nopre” means not using pre-training technique. SSLA: classification accuracy for inference task. EMA: Exact Match Accuracy for self-recovery task.

Network	SSLA	EMA
GRU-3-256-norec-nopre	89.80%	00.00%
GRU-3-256-norec-pre	91.00%	00.00%
GRU-3-256-rec-nopre	90.80%	82.25%
GRU-3-256-rec-pre	**91.77%**	**97.19%**

Table 5.10: Ablative comparison of unlabeled data pre-training and self-recovery tasks on PM2-10k data-set. ”rec” indicates the self-recovery task, ”pre” indicates the pre-training technique. Prepending ”no” means the opposite, e.g., ”nopre” means not using pre-training technique. SSLA: classification accuracy for inference task. EMA: Exact Match Accuracy for self-recovery task.

Network	SSLA	EMA
GRU-3-256-norec-nopre	50.68%	00.00%
GRU-3-256-norec-pre	63.62%	00.00%
GRU-3-256-rec-nopre	65.39%	47.23%
GRU-3-256-rec-pre	**65.40%**	**65.54%**

5.5 Conclusions

In this paper, we discuss a new semi-supervised deep learning based molecular prediction system, called **seq3seq fingerprint**. Our model is the first attempt in sequence-based deep learning method utilizing both unlabeled and labeled data for drug discovery. The reinforcement from the unlabeled data is demonstrated to significantly improve the inference performance by enhancing the representation power of the perceiver network. Furthermore, adding the auxiliary self-recovery task also augment the predictive performance. As a result, the superior inference performance over multiple state-of-the-art methods is revealed in our extensive experiments.

Our seq3seq fingerprint method still share some common aspects with Natural Language Processing (NLP) area as the seq2seq fingerprint does [10]. In the future, it might be interesting to further investigate bonds between drug discovery and NLP area, which might bring in many novel methods to further accelerate drug discovery research. The techniques described in this paper might also be extendable and beneficial to NLP research.

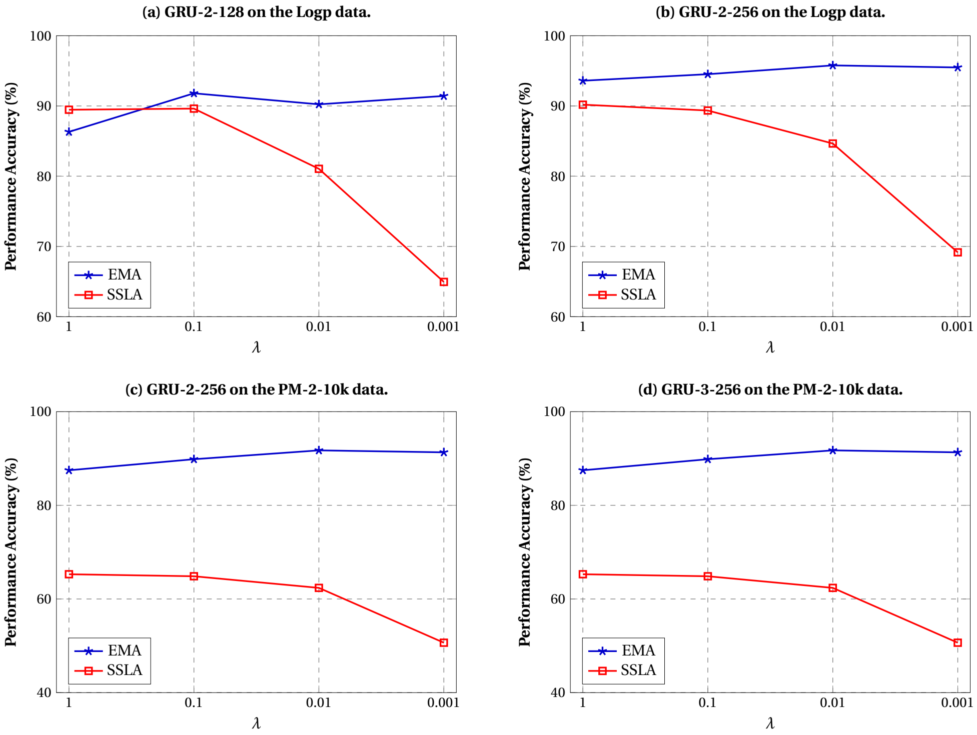


Figure 5.5: Impacts of the multi-task balance weights on different scales on both LogP and PM2-10k dataset. Within a very wide range (usually $10^{-2} - 10^{0}$), both self-recovery (EMA) and inference (SSLA) performance are quite robust to the change of λ.

CHAPTER 6

Conclusions

This thesis aims at developing large-scale deep learning techniques for large-scale data. We investigate several typical type of data in the big data era including 1) high-resolution medical images; 2) sequence drug discovery data.

We have demonstrated, both in theory and practice, our deep learning approaches formed effective and efficient solutions with clear performance gains in extensive experiments on large-scale data. Specifically, we have developed the following methods:

Efficient lung cancer cell detection with deep convolution neural network: We propose an efficient and robust lung cancer cell detection method. The proposed method is designed based on the Deep Convolution Neural Network framework[24], which is able to provide state-of-the-art accuracy with only weakly annotated ground truth. For each cell area, only one local patch containing the cell area is fed into the detector for training. The training strategy significantly reduces the time cost of training procedure due to the fact that only around one percent of all pixel labels are used. In the testing stage, by utilizing the relation of adjacent patches, the proposed method provides the exact same results within a few hundredths time. Experimental results clearly demonstrate the efficiency and effectiveness of the proposed method for large-scale lung cancer cell detection. In the future, we shall attempt to combine the structured techniques[25] to further improve the accuracy.

Detecting 10,000 cells in one second: A generalized distributed deep neural network framework is introduced to detect cells in whole-slide histopathological images. The innovative framework can be applied with any deep convolutional neural network pixel-wise cell detector. Our method is extremely optimized in distributed environment to detect cells in whole-slide images. We utilize a sparse kernel neural network forwarding technique to reduce nearly all redundant convolution computations. An asynchronous prefetching technique is recommended to diminish most disk I/O time when loading the large histopathological images into memory. Furthermore, an unbalanced distributed sampling strategy is presented to enhance the scalability and communication efficiency of our framework. These techniques construct three pillars of our framework. Extensive experiments demonstrate that our method can approximately detect $10,000$ cells per second on a single workstation, which is encouraging for high-throughput cell data.

Seq2seq fingerprint: An unsupervised deep molecular embedding for drug discovery: we discuss a new unsupervised molecular representation system, called **seq2seq fingerprint**, based on the idea from the recent breakthrough on the English-to-French language translation, named sequence to sequence learning model. Our model translates the molecular SMILE string to the SMILE itself, while at the same time generates a fixed length fingerprint vector. The experiments on classification task demonstrate its superior performance. This model translates the English sentences to French ones, but meanwhile creates a intermediate continuous vector, encoding the abstract meaning of the sentence. Our study starts from one simple question: if we translate the molecule to molecule representation itself using this model, could the intermediate vector produces a meaningful representation for future machine learning tasks? The answer is excitingly promising as demonstrated in our experiments. Also, the nature of our data-driven label-free model brings us even more benefits. 1) Our fingerprint system is completely unsupervised, meaning it will never be limited by the expensive label collection process. In fact, it could utilize each of every valid molecule, theoretically reaching the amount of infinite. 2) Contrast to the supervised learning models trained with very limited data samples, the seq2seq fingerprint is trained from a sufficiently large pool of samples, and therefore it is more robust to the specific task.

Seq3seq Fingerprint: Towards End-to-end Semi-supervised Deep Drug Discovery: We discuss a new semi-supervised deep learning based molecular prediction system, called **seq3seq fingerprint**. Our model is the first attempt in sequence-based deep learning method utilizing both unlabeled and labeled data for drug discovery. The reinforcement from the unlabeled data is demonstrated to significantly improve the inference performance by enhancing the representation power of the perceiver network. Furthermore, adding the auxiliary self-recovery task also augment the predictive performance. As a result, the superior inference performance over multiple state-of-the-art methods is revealed in our extensive experiments.

Our seq3seq fingerprint method still share some common aspects with Natural Language Processing (NLP) area as the seq2seq fingerprint does [10]. As described in Section 5.3, it looks that we have found a new direction to invent new drug discovery methods. In the future, it might be interesting to further investigate bonds between drug discovery and NLP area, which might bring in many novel methods to further accelerate drug discovery research. The techniques described in this paper might also be extendable and beneficial to NLP research.

REFERENCES

[1] C. Arteta, V. Lempitsky, J. Noble, and A. Zisserman, “Learning to detect cells using non-overlapping extremal regions,” in *Medical Image Computing and Computer-Assisted Intervention MICCAI 2012*, ser. Lecture Notes in Computer Science, N. Ayache, H. Delingette, P. Golland, and K. Mori, Eds. Springer Berlin Heidelberg, 2012, vol. 7510, pp. 348–356. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-33415-3_43

[2] Z. Xu and J. Huang, “Efficient lung cancer cell detection with deep convolution neural network,” in *International Workshop on Patch-based Techniques in Medical Imaging.* Springer, 2015, pp. 79–86.

[3] ——, “Detecting 10,000 cells in one second,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention.* Springer, 2016, pp. 676–684.

[4] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[5] E. Bernardis and X. Y. Stella, “Pop out many small structures from a very large microscopic image,” *Medical image analysis*, vol. 15, no. 5, pp. 690–707, 2011.

[6] S. Nath, K. Palaniappan, and F. Bunyak, “Cell segmentation using coupled level sets and graph-vertex coloring,” in *Medical Image Computing and Computer-Assisted Intervention MICCAI 2006*, ser. Lecture Notes in Computer Science, R. Larsen, M. Nielsen, and J. Sporring, Eds.

Springer Berlin Heidelberg, 2006, vol. 4190, pp. 101–108. [Online]. Available: http://dx.doi.org/10.1007/11866565_13

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Computer Vision and Pattern Recognition*, 2014.

[8] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[9] R. Li, W. Zhang, H.-I. Suk, L. Wang, J. Li, D. Shen, and S. Ji, "Deep learning based imaging data completion for improved brain disease diagnosis," in *Medical Image Computing and Computer-Assisted Intervention MICCAI 2014*, ser. Lecture Notes in Computer Science, P. Golland, N. Hata, C. Barillot, J. Hornegger, and R. Howe, Eds. Springer International Publishing, 2014, vol. 8675, pp. 305–312. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-10443-0_39

[10] Z. Xu, S. Wang, F. Zhu, and J. Huang, "Seq2seq fingerprint: An unsupervised deep molecular embedding for drug discovery," in *BCB*, 2017.

[11] X. Zhang, S. Wang, F. Zhu, Z. Xu, Y. Wang, and J. Huang, "Seq3seq fingerprint: Towards end-to-end semi-supervised deep drug discovery," in *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics.* ACM, 2018, pp. 404–413.

[12] T. Ching, D. S. Himmelstein, B. K. Beaulieu-Jones, A. A. Kalinin, B. T. Do, G. P. Way, E. Ferrero, P.-M. Agapow, M. Zietz, M. M. Hoffman *et al.*, "Opportunities and obstacles for deep learning in biology and medicine," *bioRxiv*, p. 142760, 2018.

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

[14] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning." in *AAAI*, vol. 4, 2017, p. 12.

[15] J. Huang and Z. Xu, "Cell detection with deep learning accelerated by sparse kernel," in *Deep Learning and Convolutional Neural Networks for Medical Image Computing.* Springer, 2017, pp. 137–157.

[16] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. van der Laak, B. van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical image analysis*, vol. 42, pp. 60–88, 2017.

[17] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, "Youtube-8m: A large-scale video classification benchmark," *arXiv preprint arXiv:1609.08675*, 2016.

[18] A. Zisserman, J. Carreira, K. Simonyan, W. Kay, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, and M. Suleyman, "The kinetics human action video dataset," 2017.

[19] J. Yao, S. Wang, X. Zhu, and J. Huang, "Imaging biomarker discovery for lung cancer survival prediction," in *International Conference on Medical Image Computing and Computer-Assisted Intervention.* Springer, 2016, pp. 649–657.

[20] F. Zhu, J. Guo, Z. Xu, P. Liao, and J. Huang, "Group-driven reinforcement learning for personalized mhealth intervention," *arXiv preprint arXiv:1708.04001*, 2017.

[21] P. Koehn, "Europarl: A parallel corpus for statistical machine translation," in *MT summit*, vol. 5, 2005, pp. 79–86.

[22] H. Li, R. Zhao, and X. Wang, "Highly efficient forward and backward propagation of convolutional neural networks for pixelwise classification," *arXiv preprint arXiv:1412.4526*, 2014.

[23] N. L. S. T. R. Team *et al.*, "The national lung screening trial: Overview and study design1," *Radiology*, 2011.

[24] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[25] J. Huang, X. Huang, and D. Metaxas, "Simultaneous image transformation and sparse representation recovery," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on.* IEEE, 2008, pp. 1–8.

[26] ——, "Learning with dynamic group sparsity," in *Computer Vision, 2009 IEEE 12th International Conference on.* IEEE, 2009, pp. 64–71.

[27] J. Huang, S. Zhang, H. Li, and D. Metaxas, "Composite splitting algorithms for convex optimization," *Computer Vision and Image Understanding*, vol. 115, no. 12, pp. 1610–1622, 2011.

[28] D. C. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Mitosis detection in breast cancer histology images with deep neural networks," in *MICCAI 2013*, K. Mori, I. Sakuma, Y. Sato, C. Barillot, and N. Navab, Eds. Springer Berlin Heidelberg, 2013, pp. 411–418.

[29] Y. Xie, F. Xing, X. Kong, H. Su, and L. Yang, "Beyond classification: Structured regression for robust cell detection using convolutional neural network," in *MICCAI 2015, Part III*, N. Navab, J. Hornegger, M. W. Wells, and F. A. Frangi, Eds. Springer International Publishing, 2015, pp. 358–365. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-24574-4_43

[30] H. Pan, Z. Xu, and J. Huang, "An effective approach for robust lung cancer cell detection," in *Patch-MI 2015*, G. Wu, P. Coupé, Y. Zhan, B. Munsell, and D. Rueckert, Eds. Springer International Publishing, 2015, pp. 87–94.

[31] A. Giusti, D. C. Cireşan, J. Masci, L. M. Gambardella, and J. Schmidhuber, "Fast image scanning with deep max-pooling convolutional neural networks," *arXiv preprint arXiv:1302.1700*, 2013.

[32] J. Mareček, P. Richtárik, and M. Takáč, "Distributed block coordinate descent for minimizing partially separable functions," in *Numerical Analysis and Optimization*. Springer, 2015, pp. 261–288.

[33] C. G. A. R. Network *et al.*, "Comprehensive molecular profiling of lung adenocarcinoma," *Nature*, vol. 511, no. 7511, pp. 543–550, 2014.

[34] J. Yao, D. Ganti, X. Luo, G. Xiao, Y. Xie, S. Yan, and J. Huang, "Computer-assisted diagnosis of lung cancer using quantitative topology features," in *Machine Learning in Medical Imaging*, L. Zhou, L. Wang, Q. Wang, and Y. Shi, Eds. Springer International Publishing, 2015, pp. 288–295.

[35] D. Edwards, "Accelerating drug development with precision dosing techniques," in *PMPS*, 2004.

[36] M. Ashton, J. Barnard, F. Casset, M. Charlton, G. Downs, D. Gorse, J. Holliday, R. Lahana, and P. Willett, "Identification of diverse database subsets using property-based and fragment-based molecular descriptions," *Molecular Informatics*, vol. 21, no. 6, pp. 598–604, 2002.

[37] G. W. Bemis and M. A. Murcko, "The properties of known drugs. 1. molecular frameworks," *Journal of medicinal chemistry*, vol. 39, no. 15, pp. 2887–2893, 1996.

[38] X. Q. Lewell, D. B. Judd, S. P. Watson, and M. M. Hann, "Recap retrosynthetic combinatorial analysis procedure: a powerful new technique for identi-

fying privileged molecular fragments with useful applications in combinatorial chemistry," *Journal of chemical information and computer sciences*, vol. 38, no. 3, pp. 511–522, 1998.

[39] S. Riniker and G. A. Landrum, "Similarity maps-a visualization strategy for molecular fingerprints and machine-learning methods," *Journal of cheminformatics*, vol. 5, no. 1, p. 43, 2013.

[40] F. Zhu, Y. Wang, S. Xiang, B. Fan, and C. Pan, "Structured sparse method for hyperspectral unmixing," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 88, pp. 101–118, 2014.

[41] F. Zhu, Y. Wang, B. Fan, S. Xiang, G. Meng, and C. Pan, "Spectral unmixing via data-guided sparsity," *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5412–5427, 2014.

[42] F. Zhu, B. Fan, X. Zhu, Y. Wang, S. Xiang, and C. Pan, "10,000+ times accelerated robust subset selection," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[43] J. Degen, C. Wegscheid-Gerlach, A. Zaliani, and M. Rarey, "On the art of compiling and using'drug-like'chemical fragment spaces," *ChemMedChem*, vol. 3, no. 10, pp. 1503–1507, 2008.

[44] L. Weber, S. Wallbaum, C. Broger, and K. Gubernator, "Optimization of the biological activity of combinatorial compound libraries by a genetic algorithm," *Angewandte Chemie International Edition in English*, vol. 34, no. 20, pp. 2280–2282, 1995.

[45] T. A. Halgren, "Merck molecular force field. iii. molecular geometries and vibrational frequencies for mmff94," *Journal of Computational Chemistry*, vol. 17, no. 5-6, pp. 553–586, 1996.

[46] J. M. Blaney and J. S. Dixon, "Distance geometry in molecular modeling," *Reviews in Computational Chemistry, Volume 5*, pp. 299–335, 2007.

[47] A. K. Rappé, C. J. Casewit, K. Colwell, W. Goddard Iii, and W. Skiff, "Uff, a full periodic table force field for molecular mechanics and molecular dynamics simulations," *Journal of the American chemical society*, vol. 114, no. 25, pp. 10 024–10 035, 1992.

[48] R. E. Carhart, D. H. Smith, and R. Venkataraghavan, "Atom pairs as molecular features in structure-activity studies: definition and applications," *Journal of Chemical Information and Computer Sciences*, vol. 25, no. 2, pp. 64–73, 1985.

[49] R. Nilakantan, N. Bauman, J. S. Dixon, and R. Venkataraghavan, "Topological torsion: a new molecular descriptor for sar applications. comparison with other descriptors," *Journal of Chemical Information and Computer Sciences*, vol. 27, no. 2, pp. 82–85, 1987.

[50] I. Wallach, M. Dzamba, and A. Heifets, "Atomnet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery," *arXiv preprint arXiv:1510.02855*, 2015.

[51] S. Kearnes, K. McCloskey, M. Berndl, V. Pande, and P. Riley, "Molecular graph convolutions: moving beyond fingerprints," *Journal of computer-aided molecular design*, vol. 30, no. 8, pp. 595–608, 2016.

[52] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," in *Advances in neural information processing systems*, 2015, pp. 2224–2232.

[53] Y. Hu, E. Lounkine, and J. Bajorath, "Improving the search performance of extended connectivity fingerprints through activity-oriented feature filtering and

application of a bit-density-dependent similarity function," *ChemMedChem*, vol. 4, no. 4, pp. 540–548, 2009.

[54] R. C. Glen, A. Bender, C. H. Arnby, L. Carlsson, S. Boyer, and J. Smith, "Circular fingerprints: flexible molecular descriptors with applications from physical chemistry to adme," *IDrugs*, vol. 9, no. 3, p. 199, 2006.

[55] H. Morgan, "The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service," *J. Chemical Documentation*, vol. 5, pp. 107–113, 1965.

[56] D. Rogers and M. Hahn, "Extended-connectivity fingerprints," *Journal of chemical information and modeling*, vol. 50, no. 5, pp. 742–754, 2010.

[57] N. M. O'Boyle, C. M. Campbell, and G. R. Hutchison, "Computational design and selection of optimal organic photovoltaic materials," *The Journal of Physical Chemistry C*, vol. 115, no. 32, pp. 16 200–16 210, 2011.

[58] C. Rupakheti, A. Virshup, W. Yang, and D. N. Beratan, "Strategy to discover diverse optimal molecules in the small molecule universe," *Journal of chemical information and modeling*, vol. 55, no. 3, pp. 529–537, 2015.

[59] O. F. Güner, *Pharmacophore perception, development, and use in drug design*. Internat'l University Line, 2000, vol. 2.

[60] G. Schneider, O. Clément-Chomienne, L. Hilfiger, P. Schneider, S. Kirsch, H.-J. Böhm, and W. Neidhart, "Virtual screening for bioactive molecules by evolutionary de novo design," *Angewandte Chemie International Edition*, vol. 39, no. 22, pp. 4130–4133, 2000.

[61] G. Schneider, M.-L. Lee, M. Stahl, and P. Schneider, "De novo design of molecular architectures by evolutionary assembly of drug-derived building blocks," *Journal of computer-aided molecular design*, vol. 14, no. 5, pp. 487–494, 2000.

[62] R. S. Pearlman and K. Smith, "Novel software tools for chemical diversity," in *3D QSAR in drug design*. Springer, 2002, pp. 339–353.

[63] F. R. Burden, "Molecular identification number for substructure searches," *Journal of Chemical Information and Computer Sciences*, vol. 29, no. 3, pp. 225–227, 1989.

[64] J. Yao, Z. Xu, X. Huang, and J. Huang, "Accelerated dynamic mri reconstruction with total variation and nuclear norm regularization," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 635–642.

[65] Z. Peng, Z. Xu, and J. Huang, "Rspirit: Robust self-consistent parallel imaging reconstruction based on generalized lasso," in *Biomedical Imaging (ISBI), 2016 IEEE 13th International Symposium on*. IEEE, 2016, pp. 318–321.

[66] Z. Xu, Y. Li, and J. Huang, "Accelerated sparse optimization for missing data completion," in *Pattern Recognition (ICPR), 2016 23rd International Conference on*. IEEE, 2016, pp. 1267–1272.

[67] Z. Xu, Y. Li, L. Axel, and J. Huang, "Effilabelscient preconditioning in joint total variation regularized parallel mri reconstruction," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 563–570.

[68] S. Wang, J. Yao, Z. Xu, and J. Huang, "Subtype cell detection with an accelerated deep convolution neural network," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2016, pp. 640–648.

[69] H. Pan, Z. Xu, and J. Huang, "An effective approach for robust lung cancer cell detection," in *International Workshop on Patch-based Techniques in Medical Imaging*. Springer, 2015, pp. 87–94.

[70] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning.* MIT Press, 2016, http://www.deeplearningbook.org.

[71] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing, and V. Pande, "Moleculenet: A benchmark for molecular machine learning," *arXiv preprint arXiv:1703.00564*, 2017.

[72] J. Gomes, B. Ramsundar, E. N. Feinberg, and V. S. Pande, "Atomic convolutional networks for predicting protein-ligand binding affinity," *arXiv preprint arXiv:1703.10603*, 2017.

[73] X. Zhu, J. Yao, F. Zhu, and J. Huang, "Wsisa: Making survival prediction from whole slide pathology images," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE, 2017.

[74] J. Yaolabels, X. Zhu, F. Zhu, and J. Huang, "Deep correlational learning for survival prediction from multi-modality data," in *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2017.

[75] D. Weininger, "Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules," in *Proc. Edinburgh Math. SOC*, vol. 17, 1970, pp. 1–14.

[76] G. Subramanian, B. Ramsundar, V. Pande, and R. A. Denny, "Computational modeling of β-secretase 1 (bace-1) inhibitors using ligand based approaches," *Journal of Chemical Information and Modeling*, vol. 56, no. 10, pp. 1936–1949, 2016.

[77] H. Altae-Tran, B. Ramsundar, A. S. Pappu, and V. Pande, "Low data drug discovery with one-shot learning," *arXiv preprint arXiv:1611.03199*, 2016.

[78] C. Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016.

[79] R. Gómez-Bombarelli, D. Duvenaud, J. M. Hernández-Lobato, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik, "Automatic chemical design using a data-driven continuous representation of molecules," *arXiv preprint arXiv:1610.02415*, 2016.

[80] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[81] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[82] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[83] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[84] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *arXiv preprint arXiv:1409.2329*, 2014.

[85] M. Abadi and et.al., "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," 2015. [Online]. Available: http://download.tensorflow.org/paper/whitepaper2015.pdf

[86] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *European conference on computational learning theory*. Springer, 1995, pp. 23–37.

[87] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[88] T. K. Ho, "Random decision forests," in *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, vol. 1. IEEE, 1995, pp. 278–282.

[89] Z. Xu and J. Huang, "A general efficient hyperparameter-free algorithm for convolutional sparse learning." in *AAAI*, 2017, pp. 2803–2809.

[90] G. Cheng, F. Zhu, S. Xiang, Y. Wang, and C. Pan, "Semisupervised hyperspectral image classification via discriminant analysis and robust regression," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 2, pp. 595–608, 2016.

[91] J. J. Irwin, T. Sterling, M. M. Mysinger, E. S. Bolstad, and R. G. Coleman, "Zinc: a free tool to discover chemistry for biology," *Journal of chemical information and modeling*, vol. 52, no. 7, pp. 1757–1768, 2012.

[92] G. Neglur, R. L. Grossman, and B. Liu, "Assigning unique keys to chemical compounds for data integration: Some interesting counter examples," in *International Workshop on Data Integration in the Life Sciences*. Springer, 2005, pp. 145–157.

[93] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing, and V. Pande, "Moleculenet: a benchmark for molecular machine learning," *Chemical Science*, vol. 9, no. 2, pp. 513–530, 2018.

[94] R. Li and J. Huang, "Learning graph while training: An evolving graph convolutional neural network," *arXiv preprint arXiv:1708.04675*, 2017.

[95] R. Li, S. Wang, F. Zhu, and J. Huang, "Adaptive graph convolutional neural networks," *arXiv preprint arXiv:1801.03226*, 2018.

[96] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," in *International conference on machine learning*, 2016, pp. 2014–2023.

[97] H. Altae-Tran, B. Ramsundar, A. S. Pappu, and V. Pande, "Low data drug discovery with one-shot learning," *ACS central science*, vol. 3, no. 4, pp. 283–293, 2017.

[98] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[99] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training."

[100] A. M. Dai and Q. V. Le, "Semi-supervised sequence learning," in *Advances in Neural Information Processing Systems*, 2015, pp. 3079–3087.

[101] T. Miyato, A. M. Dai, and I. Goodfellow, "Adversarial training methods for semi-supervised text classification," *arXiv preprint arXiv:1605.07725*, 2016.

[102] S. Swayamdipta, S. Thomson, C. Dyer, and N. A. Smith, "Frame-semantic parsing with softmax-margin segmental rnns and a syntactic scaffold," *arXiv preprint arXiv:1706.09528*, 2017.

[103] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[104] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

BIOGRAPHICAL STATEMENT

Zheng Xu received his Ph.D. in Computer Science and Engineering from the University of Texas at Arlington at 2018. Prior to beginning the Ph.D. program, Zheng obtained his B.S. degree from Huazhong University of Science and Technology, China in 2014 in Information and Computing Science. His main research interests are large-scale deep learning, medical imaging, bio-informatics and optimization techniques in sparse learning. During his Ph.D. program, he has published several papers in the top tier conferences in the literature such as the Annual International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), AAAI Conference on Artificial Intelligence (AAAI), ACM Conference on Bioinformatics, Computational Biology, and Health Informatics (ACM BCB).