TOWARDS END-TO-END SEMI-SUPERVISED DEEP LEARNING FOR DRUG DISCOVERY

by

XIAOYU ZHANG

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

MASTER OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2018

To my parents and brothers.

ACKNOWLEDGEMENTS

I'd like to express my heartfelt thanks to my supervisor, Dr. Junzhou Huang whose continual inspiration and guidance are all my life's wealth. This thesis is also dedicated to my families without whose support and encouragement, I will never make it.

I have to thank all the graduate students who gives me kind help in both academia and living life, especially Mr. Zheng Xu and Sheng Wang with who's great advice, I start my journey on deep learning. I can barely image how tough it would be to finish my thesis within such a short period without their endless guidance.

Last but not least, I would like thank all my friends, teammates and everyone who gives helpful suggestions and inspirations. I will use what I learn to contribute to the society and make more achievements in my future work.

November 8, 2018

ABSTRACT

TOWARDS END-TO-END SEMI-SUPERVISED DEEP LEARNING FOR DRUG DISCOVERY

Xiaoyu Zhang, M.S.

The University of Texas at Arlington, 2018

Supervising Professor: Dr. Junzhou Huang

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 attributed to large-scale clean high-quality labeled data, which is generally unavailable in drug discovery practices.

In this thesis, we address this issue by proposing an end-to-end deep learning framework in a semi-supervised learning fashion. That is said, the proposed deep learning approach can utilize both labeled and unlabeled data. While labeled data is of very limited availability, the amount of available unlabeled data is generally huge. The proposed framework, named as **seq3seq fingerprint**, automatically learns a strong representation of each molecule in an unsupervised way from a huge training data pool containing a mixture of both unlabeled and labeled molecules. In the meantime, the representation is also adjusted to further help predictive tasks, e.g., acidity, alkalinity or solubility classification. The entire framework is trained end-to-end and simultaneously learn the representation and inference results. Extensive experiments support the superiority of the proposed framework.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iv

ABSTRACT v

LIST OF ILLUSTRATIONS ix

LIST OF TABLES xi

Chapter Page

1. INTRODUCTION 1
    - 1.1 Neural Network and Deep Learning 1
    - 1.2 Deep Learning for drug discovery 4
    - 1.3 Problems & Challenges in Drug Discovery 4
    - 1.4 Goal of Thesis 5
2. CLASSICAL APPROACHES FOR DRUG DISCOVERY 9
    - 2.1 SMILE Representations of Molecules 9
    - 2.2 Fingerprint Methods 10
    - 2.3 Deep-learning-based Models 10
3. Methodology 12
    - 3.1 Overview 12
    - 3.2 The Duo Tasks in Seq3seq Fingerprint Model 13
    - 3.3 GRU Units 14
    - 3.4 Loss Function 15
    - 3.5 End-to-end Semi-supervised Learning 16
    - 3.6 A Multi-task Scaffolding View of Seq3seq Fingerprint 16
4. EXPERIMENTAL RESULTS 19

4.1 Experiment Setup 19

4.2 Classification Tasks 22

4.3 Regression Tasks 23

4.4 Sensitivity Analysis of Multi-task Weight Balance Parameters 26

4.5 The Ablation Study of Neural Network Structures 29

5. CONCLUSION AND FUTURE WORK 32

REFERENCES 33

BIOGRAPHICAL STATEMENT 42

LIST OF ILLUSTRATIONS

Figure Page

1.1 Architecture of artificial neural networks 2

1.2 ConvNet Architecture 3

1.3 The examples of SMILE representations. 6

a Flavopereirin: CCc(c1)ccc2[n+]1ccc3c2Nc4c3cccc4 6

b Melatonin: CC(=O)NCCC1=CNc2c1cc(OC)cc2 6

c Thiamine: OCCc1c(C)[n+](cs1)Cc2cnc(C)nc2N 6

3.1 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. 13

3.2 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. 17

4.1 Impacts of the network structures on different metrics on both LogP and PM2 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. 30

4.2 Impacts of the multi-task balance weights on different scales on both LogP and PM2 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. 31

LIST OF TABLES

Table Page

4.1 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. 23

4.2 The comparison of classification accuracy on the PM2 data. We report the average classification accuracy (Mean) and the corresponding Standard Deviation (StDev) of 5-fold cross-validation result. 23

4.3 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. 24

4.4 The comparison of 5-fold cross validation classification accuracy among different seq3seq GRU models on the PM2 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. 24

4.5 The comparison of regression results on the LogP data. We report the Root Mean Square Error(RMSE) for evaluation and the corresponding Standard Deviation (StDev) of 5-fold cross-validation result. 25

4.6 The comparison of regression results on the PM2 data. We report the Root Mean Square Error(RMSE) for evaluation and the corresponding Standard Deviation (StDev) of 5-fold cross-validation result. 25

4.7 The comparison of 5-fold cross validation regression among different seq3seq GRU models on the LogP data. Both the Root Mean Square Error(RMSE) and Standard Deviation (StDev) are reported for the 5-fold splits. FP Length: FingerPrint Length. EMA: Exact Match Accuracy for self-recovery task. 26

4.8 The comparison of 5-fold cross validation regression among different seq3seq GRU models on the PM2 data. Both the Root Mean Square Error(RMSE) and Standard Deviation (StDev) are reported for the 5-fold splits. FP Length: FingerPrint Length. EMA: Exact Match Accuracy for self-recovery task. 26

4.9 The comparison of multi-task regression among different state-of-the-art methods on the NCI data. The Root Mean Square Error(RMSE) are reported for the 5-fold splits. 27

4.10 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. 28

4.11 The performance variations with λ and GRU model parameters for PM2 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. 28

CHAPTER 1

INTRODUCTION

1.1 Neural Network and Deep Learning

Over the past decade, deep learning has achieved significant success in various areas. The most impactful event is AlphaGo knocked out Lee Sedol in 2016. Actually, deep learning has already begin to change our daily life, such as self-driving car and video surveillance. There are multiple reasons why Artificial intelligence is so successfully applied into modern industrial manufacturing. First, a huge amount of high quality data is available from multiple resources, which makes the complex model training available in order to give accurate enough predictions. Today, the information from internet are growing exponentially every year, a huge amount of data are being collected from almost all the aspects of human life and saved to economic storage with large capacity. Secondly, the appearance of computational powerful CPUs and GPUs make the large scale computation affordable for the industry. The tensor processing unit(TPU) and many GPU friendly software packages such as Tensorflow[1], Pytorch[37] and Caffe[24] are available for research purpose and industrial production. All these make deep learning become practical. Obviously, deep learning related technologies have and will continue to benefit our daily life, especially in the era of big data.

The deep learning algorithm is based on neural network that non-linearly combine a number of layers as shown in 1.1, each layer contains a number of connected neurons, each neurons also connects with neurons within the neighboring layers. This kind of architecture is all artificial neural network(ANN). The combination of all the

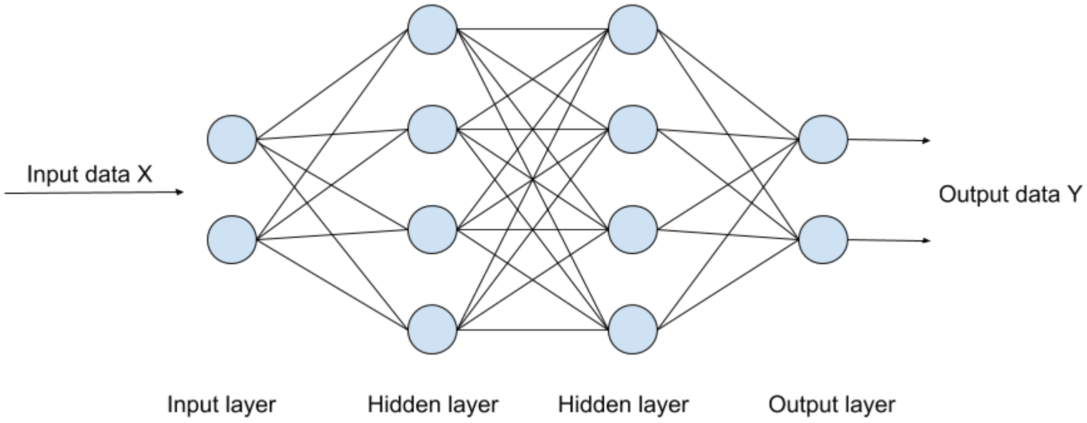


Figure 1.1: Architecture of artificial neural networks

layers of ANN can finally extract high-level features and model the nonlinear propagation of input data with the help of activation function g in 1.1. By adjusting the weights according to the data, non-linear model can be updated gradually and we use the converged solution to classify the data with discrete label or predicting properties with continuous number.

$$Y_i = g\Big\langle \sum_j W_{ij} * a_j \Big\rangle \tag{1.1}$$

where a_j refers to input variables, W is the weight matrix, Y is output, and g is an activation function which can be rectified linear unit(ReLU), Sigmoid, Hyper Tangent or even Linear.

Convolution neural network(CNN)1.2[1] has become the most popular neural network in the past decade, especially in image recognition. Usually, it contains many inceptions that made up of some various types of layers in one neural network,

[1] https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/

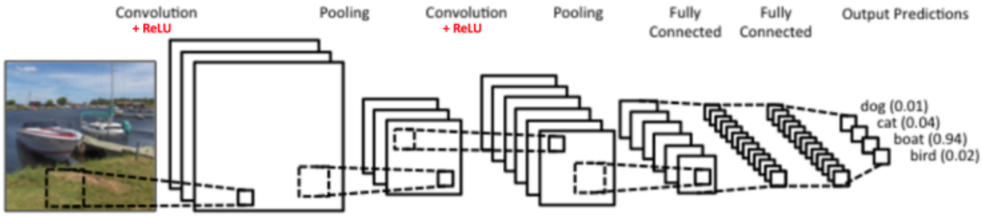


Figure 1.2: ConvNet Architecture

which includes convolution layers, pooling layers, drop out layers and fully connected layers. The convolution layers use a set of filters, which greatly reduce the number of intermediate parameters since they share the same weights. In addition, the pooling layers re-sample previous layers and decrease the size of intermediate layers. Thus, CCN is much more efficient compared with the traditional artificial neural network. Besides, the drop out layers can suppress over-fitting that is inherited from traditional artificial neural network. Many detection or imaging problems can be solved by deep learning with CNN, such as image recognition[39, 15], medical imaging[51, 68, 67, 48, 58, 21, 55, 57] and seismic imaging[65, 63, 64].

Recurrent Neural Network(RNN) is widely used in Nature Language Processing (NLP) [12]. Unlike feedforward neural networks, the output of RNN not only depends on the current inputs but also the output of previous node within the same layer as shown by 1.2. It's usually used for sequence model to keep sequential features. Assume that RNN computes sequence of outputs $(y_1, \ldots, y_T)$ from the input sequences $(x_1, \ldots, x_T)$ by iterating This type of NN can be used to process time-serious signals since it holds hidden memory that keeps the sequential features of the inputs. One well known variant is called Long short memory(LSTM)[19]. There are many successful applications of RNN in various areas such as object detection[60], action recognition, text recognition, and drug discovery[11, 56, 62].

$$
\begin{aligned}
a^{\langle t \rangle} &= g\Big(W_{aa}a^{\langle t-1 \rangle} + W_{ax}x^{\langle t \rangle} + b_a\Big) \\
y^{\langle t \rangle} &= g\Big((W_{ya}a^{\langle t \rangle} + b_y\Big)
\end{aligned}
\tag{1.2}
$$

where W_{aa} is the weight for activation a^{t-1}, W_{ax} is the weight for inputsx^t, g is the activation function.

1.2 Deep Learning for drug discovery

Recently, the application of some advanced Artificial Intelligence(AI) technologies in drug discovery has become significant and increasingly popular[6, 25]. 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 [7]. In fact, most of NN models have been applied to drug discovery, such as CNN[46], RNN[59, 61] and Generative Deep Neural Network and reinforcement learning technology[23, 29]. RNN has the advantage of processing sequential models, which is suitable for drug analysis. It's not easy to train GANs and the reinforcement learning model due to model collapse issue, researchers are currently making a lot of efforts to optimize the model and improve those technology for drug discovery.

1.3 Problems & Challenges in Drug Discovery

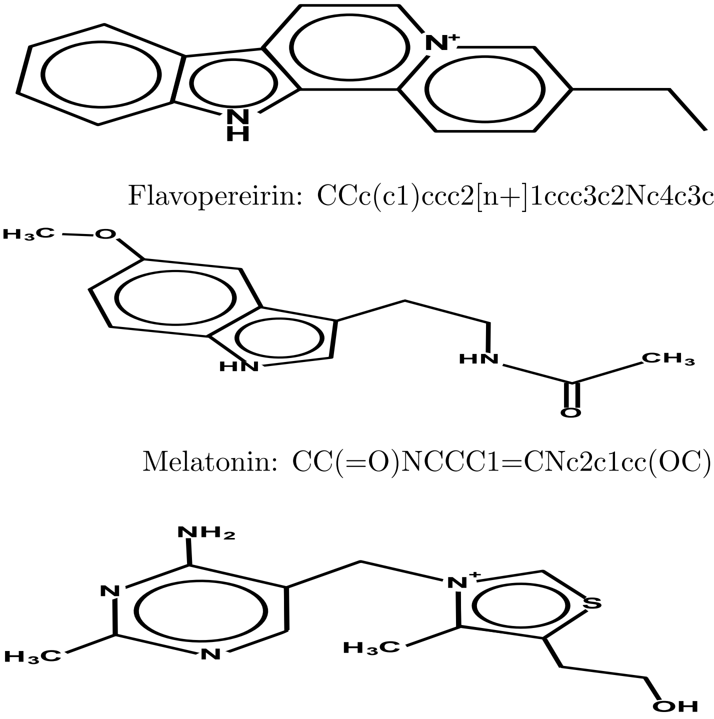
The previous success of deep learning in multiple applications, e.g., image understanding [10, 45], medical imaging [21, 52, 32, 48], video understanding [2, 69], bioinformatics [56, 58, 66], and machine translation [27], etc., has implied a reliance on large-scale high-quality labeled data-sets. The 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 [36]. 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 [22] 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.

1.4 Goal of Thesis

In this thesis, 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** [56], 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 seq3seq fingerprint framework contains **three** ends with one input and two outputs while the seq2seq fingerprint contains **two** ends with one input and one output.

Figure 1.3: The examples of SMILE representations.



Flavopereirin: CCc(c1)ccc2[n+]1ccc3c2Nc4c3cccc4

Melatonin: CC(=O)NCCC1=CNc2c1cc(OC)cc2

Thiamine: OCCc1c(C)[n+](cs1)Cc2cnc(C)nc2N

To briefly introduce the proposed seq3seq fingerprint framework[62], 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 1.3. 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.

The technical contributions of this thesis 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) several important features are enabled in the seq3seq fingerprint to help inference:

- this is the first **end-to-end** framework coupling both the recovery and inference task.
- the proposed framework is general enough to suit **different prediction tasks**, e.g., classification, regression, etc.
- it is feasible to use **different inference network structures**, e.g., Convolutional Neural Networks (CNNs), Multi-Layer Perceptrons (MLPs), etc.

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 2. In Section 3, we describe our entire pipeline in details. We show our experiment results in Section 4, demonstrating the superior performance of our method. We conclude and discuss the future direction of our paper in Chapter 5.

CHAPTER 2

CLASSICAL APPROACHES FOR DRUG DISCOVERY

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 [5], seq2seq fingerprint [56].

2.1 SMILE Representations of Molecules

Initially, the molecules are stored in the form of a sequence representation, namely the Simplified Molecular-Input Line-Entry system (SMILE) [49], 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 1.3.

2.2 Fingerprint Methods

1. **Hash-based Fingerprints** Many hash-based methods has been developed to generate unique molecular feature representation [20, 16, 33]. 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) [40]. However, due to the non-invertable 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.
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., [35, 41]. 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.

2.3 Deep-learning-based Models

The growth of deep learning [51, 28] has provided the great flexibility and performance to create the molecular fingerprint from data samples, without explicit human guide, [11, 47, 26, 42, 3, 56]. In this subsection, we discuss two major classes, namely supervised and unsupervised learning models.

1. **Supervised Models**

Many of deep learning-based fingerprint methods are still trained in a supervised-learning fashion [42, 50], which is using only labeled molecular data samples as inputs and adjusting model weights according to their labels [30]. 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 [11]. 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 convolution neural network [17, 31, 34, 30]. There are also few attempts that address the insufficient label issue by using few-shot learning strategies, e.g., [4]. 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.

2. **Unsupervised Models**

   Recently, few unsupervised fingerprint methods, e.g., seq2seq fingerprint [56], 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 [13], GradientBoost [14], and RandomForest [18], 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.

CHAPTER 3

Methodology

3.1 Overview

In this chapter, we describe the details of our semi-supervised seq3seq fingerprint model[62]. 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). After that, we describe the recovery task and the inference task in detail, their loss functions and how the two tasks are trained. Then the semi-supervised loss is described. In the end, we offer a multi-task scaffolding view from frame-semantic parsing [44] in natural language processing area to explain the proposed model.

Different from traditional models [5, 56], 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 3.1. The semi-supervised training is done by two tasks: the self-recovery task and the inference task. The whole pipeline is illustrated in Figure 3.2.

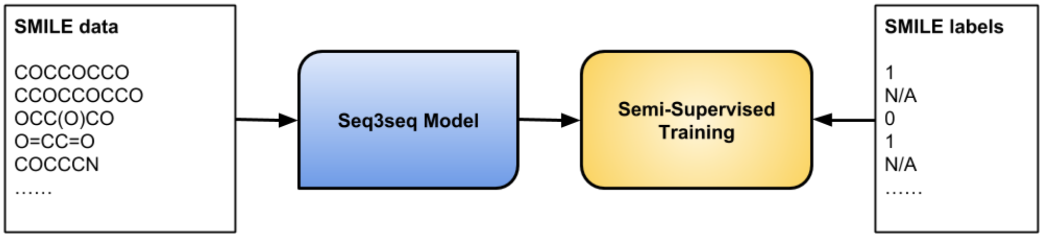


Figure 3.1: This figures shows how semi-supervised training is used for our proposed model. We mix the unlabeled data and labeled data together to train our proposed model. The SMILEs with label 0/1 come from labeled dataset and the SMILEs without labels (N/A in the figure) come from unlabeled dataset.

3.2 The Duo Tasks in Seq3seq Fingerprint Model

The Self-recovery Task The self-recovery task is to learn a vector representation (usually noted as **fingerprint** in the drug discovery literature) for each input molecular SMILE string. This task also requires the SMILE string of the molecule can be recovered from its fingerprint vector. It is an unsupervised learning problem since no label information is required in training. As shown in Figure 3.2, this task contains a perceiver network and an interpreter network. This structure is motivated by the seq2seq model [56, 43]. The original seq2seq model is used in machine translation [43]. 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 [56] combines the idea from seq2seq learning and the idea of auto-encoder to learn the vector representation for molecule.

We generalize the idea of seq2seq [5, 56] in two views. First, the perceiver network and the interpreter network in the proposed seq3seq fingerprint model can be any recurrent deep neural networks such as LSTM, GRU neural networks. The only limitation is that the perceiver network could map the string tokens into a vector representation and the interpreter could map the vector back into string tokens.

Second, we introduce unlabeled molecule data into our training process to learn better representations. Instead of using the SMILE strings of only the labeled molecule data, we take advantage of the **almost infinite** unlabeled data and use both unlabeled and labeled data for the self-recovery task to learn a more accurate vector presentation than those models which only use labeled data or unlabeled data separately. The loss function in our proposed model follows the one in [56]. It is the sum of multiple cross-entropy loss and we denote it as $\mathcal{L}_{unsup}$.

The 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 Convolution 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.

3.3 GRU Units

We implement our method with the Gated Recurrent Unit (GRU) though, out algorithm can be generalized to take advantage of other RNN units such as LSTM.

Given a sequence of input sequences $(x_1, \ldots, x_T)$, the outputs $(y_1, \ldots, y_T)$ can be calculated by:

$$\begin{aligned}
z_t &= \sigma_g(W_z x_t + U_z y_{t-1} + b_z) \\
r_t &= \sigma_r(W_r x_t + U_r y_{t-1} + b_r) \\
h_t &= \tanh(U_h x_t + W_h(y_{t-1} \circ r_t)) \\
y_t &= (1 - z_t) \circ h_{t-1} + z_t \circ y_{t-1}. && (3.1)
\end{aligned}$$

Two gates are used, one is the update gate z, the other is the reset gate r. W, U, b can be gradually adjusted in the learning process for both of the two gates . We often choose sigmoid function as the transfer function σ. GRU use a smaller number of parameters so it's faster than LSTM, the performance is comparable though[9].

3.4 Loss Function

The cross-entropy loss is used in our classification experiments, the inputs of which are predictions and labels. It calculates the probability error for the classification task which requires each data point is exclusively belong to one class3.2. Specifically, it's the negative log-likelihood of true label given predictions. For binary classification, given true label $y \in 0, 1$, and estimated probability p=Pr(y=1), the log loss per sample is the negative likelihood given the true label:

$$L_log(y, p) = -LogPr(y|p) = -(ylog(p) + (1 - y)log(1 - p)) \tag{3.2}$$

Another loss function for regression is Mean Square Error(MSE)3.3, which computes the mean of the sum of squares errors between continuous predictions and the corresponding labels. It a measure of how predictions are close to real value targets. MSE and its variant are robust and widely used in machine learning algorithm.

$$MSE(y, \hat{y}) = \frac{1}{n_{sample}} \sum_{i=0}^{n_{sample}-1} (y - \hat{y})^2 \tag{3.3}$$

where y denote labels, $\hat{y}$ denote predictions, n_{sample} is the number of samples.

3.5 End-to-end Semi-supervised Learning

As shown in Figure 3.2, the semi-supervised loss $\mathcal{L}_{semi}$ combines the unsupervised loss $\mathcal{L}_{unsup}$ and the supervised loss $\mathcal{L}_{sup}$ together as

$$\mathcal{L}_{semi} = \mathcal{L}_{unsup} + \lambda \mathcal{L}_{sup}. \tag{3.4}$$

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. 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 multi-stage training, i.e., pre-trained model training or separated classifier training [56]. As a result, the proposed end-to-end model is expected to provide an optimal inference performance as well as shorter training time for specific task than that in a multi-stage model from [56].

3.6 A Multi-task Scaffolding View of Seq3seq Fingerprint

In [56], 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** [44] in the NLP area as well. In [44], 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

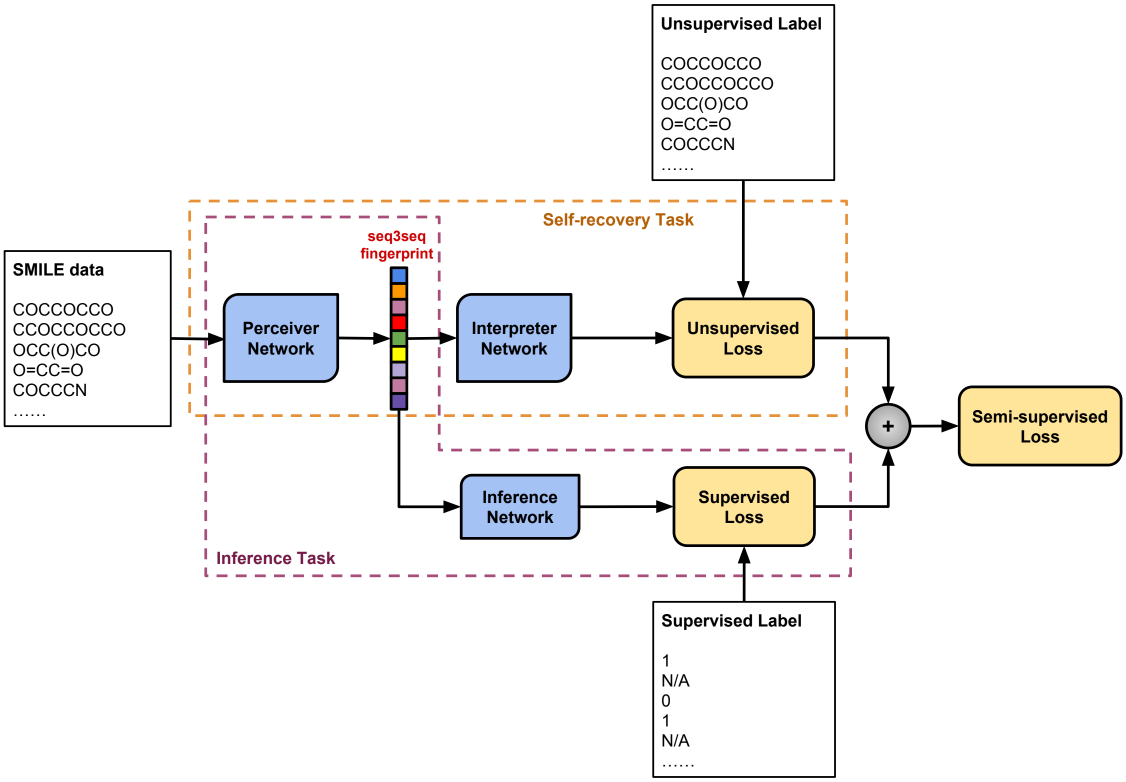


Figure 3.2: This figure shows the proposed seq3seq fingerprint model. The proposed model is trained through two tasks: a self-recovery task and an inference task. The self-recovery task contains a perceiver network and an interpreter network; the inference task shares the perceiver with self-recover task and has an inference network. The semi-supervised loss is the sum of supervised loss and unsupervised loss.

sequence-to-frame network model generally performs poorly in this task. In [44], 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 [44], 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 Chapter 4.

CHAPTER 4

EXPERIMENTAL RESULTS

In this chapter, 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, both classification and regression are implemented and improvements valid our model. 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, etc.

4.1 Experiment Setup

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 [22]. 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, 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.

- **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**: PM2 dataset contains 200,000 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.
- **NCI**: the data collections are produced by major NCI initiatives and other widely used datasets. The dataset we use contains about 19127 unique samples of SMILE strings and corresponding continuous float labels. We use eight properties for our experiments

  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 only discuss Gated Recurrent Unit (GRU) [8] as the RNN cell. 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, $L \in N^+$ is the stacked RNN layer number and $H \in N^+$ 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 [56] as well as several other state-of-the-art methods: the ECFP [40] (circular fingerprint) and the neural fingerprint method [11]. We download the official implementation of the seq2seq fingerprint [1] and carefully follow the experimental setting of the authors. The circular fingerprint is a hand-crafted hash-based feature that was generated through RDKit [2]. The neural fingerprint

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

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

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 [1], 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.

4.2 Classification Tasks

In Table 4.1 and 4.2, we report the 5-fold cross validation average classification accuracy on LogP and PM2 datasets. The proposed methods are compared with ECFP (circular) fingerprint [16], neural fingerprint [5] and seq2seq fingerprint [56]. 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 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 4.1: 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 [40]	Neural [11]	seq2seq [56]	seq3seq (Ours)
Mean	36.74%	60.80%	76.64%	**89.72%**
StDev	0.74%	1.35%	0.43%	0.41%

Table 4.2: The comparison of classification accuracy on the PM2 data. We report the average classification accuracy (Mean) and the corresponding Standard Deviation (StDev) of 5-fold cross-validation result.

	Circular [40]	Neural [11]	seq2seq [56]	seq3seq (Ours)
Mean	39.38%	52.27%	62.06%	**68.45%**
StDev	1.14%	1.12%	1.98%	0.80%

4.3 Regression Tasks

We utilize LogP and PM2 data to show the validness of our model. In Table 4.5 and 4.6, the 5-fold cross validation average RMSE on LogP and PM2 datasets were shown. It's obvious that our model, seq3seq fingerprint, provides the smallest RMSE on both datasets. Notice that the RMSE with our method has decreased by about 55% compared with ECFP (circular) fingerprint [16], and about 50% compared with neural fingerprint [5] and seq2seq fingerprint [56] for LogP data. This is because

Table 4.3: 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 4.4: The comparison of 5-fold cross validation classification accuracy among different seq3seq GRU models on the PM2 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%

circular fingerprint generate each layer's feature by applying a fixed hash function to the features of the neighborhood in the previous layer, which doesn't encode complete information, and seq2seq fingerprint can't take advantage of labeled data. The way neural fingerprint generating fingerprint just mimic circular fingerprint which also can not utilize all the information available. Unfortunately, a large amount of unlabeled data is not utilizable for neural fingerprint. Similarly, our method still outperforms the other state-of-the-art methods on PM2 dataset, which further demonstrate the validness of our method.

Table 4.7 and 4.8 show the comparisons of 5-fold cross validation regression among differen seq3seq GRU models on LogP and PM2 data. As expected, larger size of GRU units don't always give us better performance. For both of two datasets, GRU-4-128 gives us smallest RMSE which is consistent with classification results.

Table 4.5: The comparison of regression results on the LogP data. We report the Root Mean Square Error(RMSE) for evaluation and the corresponding Standard Deviation (StDev) of 5-fold cross-validation result.

	Circular [40]	Neural [11]	seq2seq [56]	seq3seq (Ours)
Mean	1.223	1.149	1.047	**0.5399**
StDev	0.0076	0.0133	0.0041	0.0043

Table 4.6: The comparison of regression results on the PM2 data. We report the Root Mean Square Error(RMSE) for evaluation and the corresponding Standard Deviation (StDev) of 5-fold cross-validation result.

	Circular [40]	Neural [11]	seq2seq [56]	seq3seq (Ours)
Mean	0.0944	0.0887	0.0808	**0.0535**
StDev	0.0026	0.0024	0.0029	0.0010

For multi-task regression, we utilize 19127 SMILEs and 8 properties from NCI data. Simillarly, the prediction results overperform all the other comparison method. One advantage of multi-task regression task is that the predictions for the 8 properties were got with one training process, instead of training separately. A second benefit which should be noticed is the model converges faster by training 8 properties simultaneously compared with training each task separately. For some properties, the final result would be also slightly better than sigle task learning.

Table 4.7: The comparison of 5-fold cross validation regression among different seq3seq GRU models on the LogP data. Both the Root Mean Square Error(RMSE) and Standard Deviation (StDev) are reported for the 5-fold splits. FP Length: FingerPrint Length. EMA: Exact Match Accuracy for self-recovery task.

	GRU-3-128	GRU-4-128	GRU-5-128	GRU-3-256	GRU-4-256	GRU-5-256
FP Length	384	512	640	768	1024	1280
RMSE Mean	0.5659	**0.5399**	0.5998	0.5470	0.5482	0.5945
RMSE StDev	0.0050	0.0075	0.0055	0.0043	0.0039	0.0038
EMA Mean	0.6778	0.6760	0.6743	0.6703	0.6831	0.6346
EMA StDev	0.0120	0.0041	0.0047	0.0036	0.0041	0.0036

Table 4.8: The comparison of 5-fold cross validation regression among different seq3seq GRU models on the PM2 data. Both the Root Mean Square Error(RMSE) and Standard Deviation (StDev) are reported for the 5-fold splits. FP Length: FingerPrint Length. EMA: Exact Match Accuracy for self-recovery task.

	GRU-3-128	GRU-4-128	GRU-5-128	GRU-2-256	GRU-3-256	GRU-4-256
FP Length	384	512	640	512	768	1024
RMSE Mean	0.0571	**0.0535**	0.0541	0.0551	0.0547	0.0542
RMSE StDev	0.0023	0.0010	0.0013	0.0014	0.0047	0.0010
EMA Mean	0.8574	0.8969	0.8689	0.8914	0.9046	0.8941
EMA StDev	0.0094	0.0159	0.0348	0.0073	0.0054	0.0034

4.4 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 loss function) 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 section, 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.

The balance weight λ control the learning process so we have to answer how it affect the accuracy performance. Increasing λ value enhances the weight of the supervised learning takes in the total loss functionand vice-versa. A large λ value does't

Table 4.9: The comparison of multi-task regression among different state-of-the-art methods on the NCI data. The Root Mean Square Error(RMSE) are reported for the 5-fold splits.

	CCRF	HL-60	K-562	RPMI	A549	COLO	HCC	MALME
Seq3seq(Ours)	**0.790**	**0.925**	**0.934**	**0.805**	**1.030**	**1.021**	**0.048**	**0.864**
Seq2seq [56]	1.512	1.771	1.787	1.641	1.771	1.754	1.623	1.654
Neural[5]	1.659	1.843	1.862	1.791	1.963	1.914	1.881	1.815
Circular[40]	1.766	1.968	1.988	1.901	2.088	2.068	1.895	1.931

necessarily to generate the best fit models in our experiments since it's an optimization problem how to combine the loss of the unsupervised learning and supervised learning with a penalty *lambda*; We train the model with λ values ranging from 1 to 0.001. Unsupervised learning and supervised learn interact with each other to get the minimum total loss. Since these two learning module share the same weights of the seq3seq neural network, the SSL and EM accuracy increasing trend also varies with λ. Among those values, λ value 1(accuracy 0.8631) has the worst accuracy performance; so $\lambda = 1$ is probably not the right choice. The larger the λ value, the earlier the accuracy of supervised learning start to increase. However, the final EMA and SSLA results vary little when λ varies within a large range.

In Table 4.10, 4.11 as well as Figure 4.2, 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.

Table 4.10: 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%

Table 4.11: The performance variations with λ and GRU model parameters for PM2 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%

4.5 The Ablation Study of Neural Network Structures

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 4.3 and 4.4 for each of the two datasets, respectively. Figure 4.1 (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 4.1, 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.

There's no guarantee that highest EM accuracy seq3seq fingerprint model always results in the best classification accuracy. Longer fingerprints might contain more information. However, classification accuracy doesn't increase with EM accuracy monotonically due to multiple reasons. First, in order to get high EM accuracy,

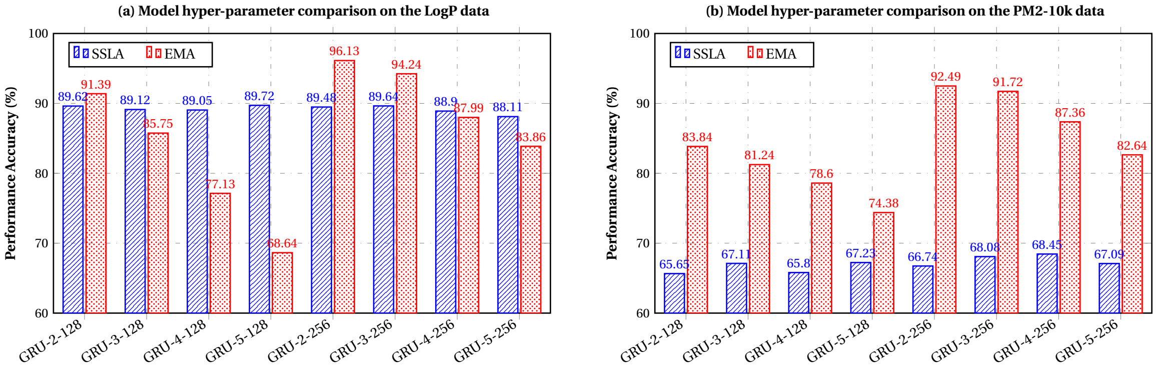
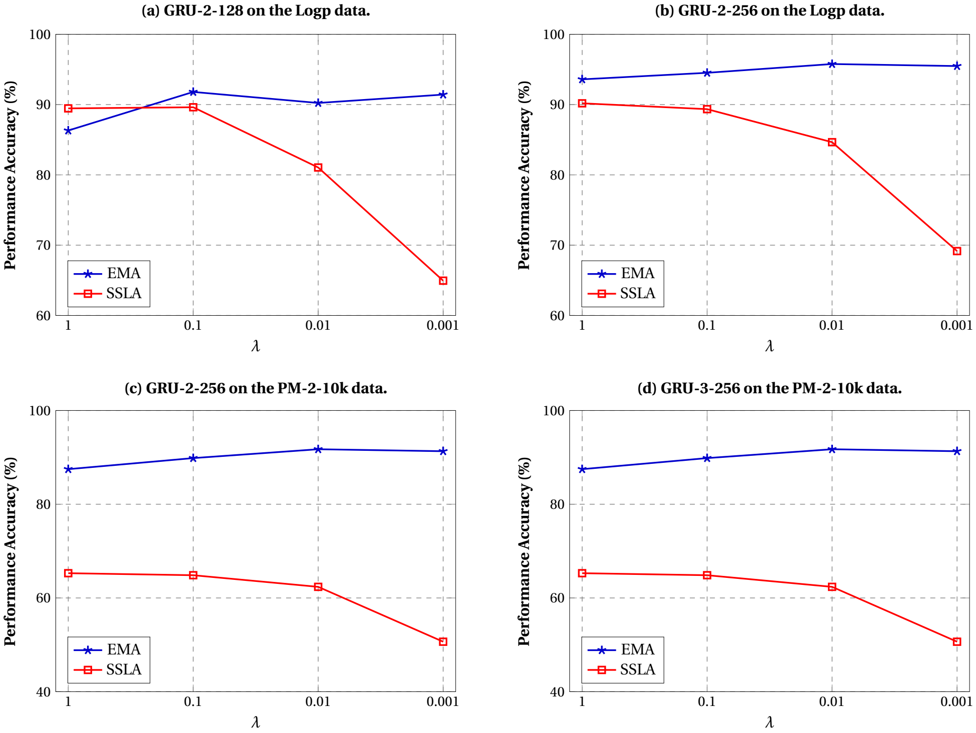


Figure 4.1: Impacts of the network structures on different metrics on both LogP and PM2 dataset. 1) The robustness of inference performance (SSLA, blue bars) is revealed. 2) The positive and negative correlations with regard to the self-recovery performance (EMA, red bars) are observed for RNN network depths and widths, respectively.

longer fingerprint takes much more time to train than small sized fingerprints. Secondly, it's easy for longer fingerprint to bring in noise. In consequence, seq3seq model GRU-5-128 on LogP data has higher SSLA but not EM accuracy than other even smaller model for example GRU-4-128. Similarly, PM2 testing results shows seq3seq-1024 has higher SSLA than seq3seq-768, but not EM accuracy.


(a) GRU-2-128 on the Logp data.

(b) GRU-2-256 on the Logp data.

(c) GRU-2-256 on the PM-2-10k data.

(d) GRU-3-256 on the PM-2-10k data.

Figure 4.2: Impacts of the multi-task balance weights on different scales on both LogP and PM2 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 5

CONCLUSION AND FUTURE WORK

In this thesis, 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. As a result, the superior inference performance over multiple state-of-the-art methods is revealed in our extensive experiments.

In the future, a potential direction might be improving the training algorithm [53, 38, 54]. Furthermore, our seq3seq fingerprint method still share some common aspects with Natural Language Processing (NLP) area as the seq2seq fingerprint does [56]. As described in Section 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.

REFERENCES

[1] Martn Abadi and et.al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems, 2015.

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

[3] Han Altae-Tran, Bharath Ramsundar, Aneesh S Pappu, and Vijay Pande. Low data drug discovery with one-shot learning. *arXiv preprint arXiv:1611.03199*, 2016.

[4] Han Altae-Tran, Bharath Ramsundar, Aneesh S Pappu, and Vijay Pande. Low data drug discovery with one-shot learning. *ACS central science*, 3(4):283–293, 2017.

[5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[6] Hongming Chen, Ola Engkvist, Yinhai Wang, Marcus Olivecrona, and Thomas Blaschke. The rise of deep learning in drug discovery. *Drug discovery today*, 2018.

[7] Travers Ching, Daniel S Himmelstein, Brett K Beaulieu-Jones, Alexandr A Kalinin, Brian T Do, Gregory P Way, Enrico Ferrero, Paul-Michael Agapow, Michael Zietz, Michael M Hoffman, et al. Opportunities and obstacles for deep learning in biology and medicine. *bioRxiv*, page 142760, 2018.

[8] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[9] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[10] 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.

[11] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pages 2224–2232, 2015.

[12] Santiago Fernández, Alex Graves, and Jürgen Schmidhuber. An application of recurrent neural networks to discriminative keyword spotting. In *International Conference on Artificial Neural Networks*, pages 220–229. Springer, 2007.

[13] Yoav Freund and Robert E Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer, 1995.

[14] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[15] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[16] Robert C Glen, Andreas Bender, Catrin H Arnby, Lars Carlsson, Scott Boyer, and James Smith. Circular fingerprints: flexible molecular descriptors with applications from physical chemistry to adme. *IDrugs*, 9(3):199, 2006.

[17] Joseph Gomes, Bharath Ramsundar, Evan N Feinberg, and Vijay S Pande. Atomic convolutional networks for predicting protein-ligand binding affinity. *arXiv preprint arXiv:1703.10603*, 2017.

[18] Tin Kam Ho. Random decision forests. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, volume 1, pages 278–282. IEEE, 1995.

[19] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[20] Ye Hu, Eugen Lounkine, and Jürgen Bajorath. Improving the search performance of extended connectivity fingerprints through activity-oriented feature filtering and application of a bit-density-dependent similarity function. *ChemMedChem*, 4(4):540–548, 2009.

[21] Junzhou Huang and Zheng Xu. Cell detection with deep learning accelerated by sparse kernel. In *Deep Learning and Convolutional Neural Networks for Medical Image Computing*, pages 137–157. Springer, 2017.

[22] John J Irwin, Teague Sterling, Michael M Mysinger, Erin S Bolstad, and Ryan G Coleman. Zinc: a free tool to discover chemistry for biology. *Journal of chemical information and modeling*, 52(7):1757–1768, 2012.

[23] Natasha Jaques, Shixiang Gu, Dzmitry Bahdanau, José Miguel Hernández-Lobato, Richard E Turner, and Douglas Eck. Sequence tutor: Conservative fine-tuning of sequence generation models with kl-control. *arXiv preprint arXiv:1611.02796*, 2016.

[24] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[25] Yankang Jing, Yuemin Bian, Ziheng Hu, Lirong Wang, and Xiang-Qun Sean Xie. Deep learning for drug design: An artificial intelligence paradigm for drug discovery in the big data era. *The AAPS journal*, 20(3):58, 2018.

[26] Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design*, 30(8):595–608, 2016.

[27] Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86, 2005.

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

[29] Albert Leo, Corwin Hansch, and David Elkins. Partition coefficients and their uses. *Chemical reviews*, 71(6):525–616, 1971.

[30] Ruoyu Li and Junzhou Huang. Learning graph while training: An evolving graph convolutional neural network. *arXiv preprint arXiv:1708.04675*, 2017.

[31] Ruoyu Li, Sheng Wang, Feiyun Zhu, and Junzhou Huang. Adaptive graph convolutional neural networks. *arXiv preprint arXiv:1801.03226*, 2018.

[32] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen AWM van der Laak, Bram van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

[33] HL Morgan. The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service. *J. Chemical Documentation*, 5:107–113, 1965.

[34] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *International conference on machine learning*, pages 2014–2023, 2016.

[35] Noel M O'Boyle, Casey M Campbell, and Geoffrey R Hutchison. Computational design and selection of optimal organic photovoltaic materials. *The Journal of Physical Chemistry C*, 115(32):16200–16210, 2011.

[36] Hao Pan, Zheng Xu, and Junzhou Huang. An effective approach for robust lung cancer cell detection. In *International Workshop on Patch-based Techniques in Medical Imaging*, pages 87–94. Springer, 2015.

[37] Adam Paszke, Sam Gross, Soumith Chintala, and Gregory Chanan. Pytorch, 2017.

[38] Zhongxing Peng, Zheng Xu, and Junzhou Huang. Rspirit: Robust self-consistent parallel imaging reconstruction based on generalized lasso. In *Biomedical Imaging (ISBI), 2016 IEEE 13th International Symposium on*, pages 318–321. IEEE, 2016.

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

[40] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50(5):742–754, 2010.

[41] Chetan Rupakheti, Aaron Virshup, Weitao Yang, and David N Beratan. Strategy to discover diverse optimal molecules in the small molecule universe. *Journal of chemical information and modeling*, 55(3):529–537, 2015.

[42] Govindan Subramanian, Bharath Ramsundar, Vijay Pande, and Rajiah Aldrin Denny. Computational modeling of β-secretase 1 (bace-1) inhibitors using ligand based approaches. *Journal of Chemical Information and Modeling*, 56(10):1936–1949, 2016.

[43] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[44] Swabha Swayamdipta, Sam Thomson, Chris Dyer, and Noah A Smith. Frame-semantic parsing with softmax-margin segmental rnns and a syntactic scaffold. *arXiv preprint arXiv:1706.09528*, 2017.

[45] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017.

[46] Thomas Unterthiner, Andreas Mayr, Günter Klambauer, and Sepp Hochreiter. Toxicity prediction using deep learning. *arXiv preprint arXiv:1503.01445*, 2015.

[47] Izhar Wallach, Michael Dzamba, and Abraham Heifets. Atomnet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery. *arXiv preprint arXiv:1510.02855*, 2015.

[48] Sheng Wang, Jiawen Yao, Zheng Xu, and Junzhou Huang. Subtype cell detection with an accelerated deep convolution neural network. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 640–648. Springer, 2016.

[49] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. In *Proc. Edinburgh Math. SOC*, volume 17, pages 1–14, 1970.

[50] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical Science*, 9(2):513–530, 2018.

[51] Zheng Xu and Junzhou Huang. Efficient lung cancer cell detection with deep convolution neural network. In *International Workshop on Patch-based Techniques in Medical Imaging*, pages 79–86. Springer, 2015.

[52] Zheng Xu and Junzhou Huang. Detecting 10,000 cells in one second. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 676–684. Springer, 2016.

[53] Zheng Xu and Junzhou Huang. A general efficient hyperparameter-free algorithm for convolutional sparse learning. In *AAAI*, pages 2803–2809, 2017.

[54] Zheng Xu, Yeqing Li, Leon Axel, and Junzhou Huang. Efficient preconditioning in joint total variation regularized parallel mri reconstruction. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 563–570. Springer, 2015.

[55] Zheng Xu, Sheng Wang, Yeqing Li, Feiyun Zhu, and Junzhou Huang. Prim: An efficient preconditioning iterative reweighted least squares method for parallel brain mri reconstruction. *Neuroinformatics*, pages 1–6, 2018.

[56] Zheng Xu, Sheng Wang, Feiyun Zhu, and Junzhou Huang. Seq2seq fingerprint: An unsupervised deep molecular embedding for drug discovery. In *BCB*, 2017.

[57] Chaochao Yan, Jiawen Yao, Ruoyu Li, Zheng Xu, and Junzhou Huang. Weakly supervised deep learning for thoracic disease classification and localization on chest x-rays. In *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 103–110. ACM, 2018.

[58] Jiawen Yao, Sheng Wang, Xinliang Zhu, and Junzhou Huang. Imaging biomarker discovery for lung cancer survival prediction. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 649–657. Springer, 2016.

[59] JC Yarrow, Y Feng, ZE Perlman, T Kirchhausen, and TJ Mitchison. Phenotypic screening of small molecule libraries by high throughput cell imaging. *Combinatorial chemistry & high throughput screening*, 6(4):279–286, 2003.

[60] Hu Han Shiguang Shan Yu Song, Yuanshun Cui and Xilin Chen. Scene text detection via deep semantic feature fusion and attention-based refinement. 2018.

[61] William Yuan, Dadi Jiang, Dhanya K Nambiar, Lydia P Liew, Michael P Hay, Joshua Bloomstein, Peter Lu, Brandon Turner, Quynh-Thu Le, Robert Tibshirani, et al. Chemical space mimicry for drug discovery. *Journal of chemical information and modeling*, 57(4):875–882, 2017.

[62] Xiaoyu Zhang, Sheng Wang, Feiyun Zhu, Zheng Xu, Yuhong Wang, and Junzhou 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*, pages 404–413. ACM, 2018.

[63] Xiaoyu Zhang, Qunshan Zhang*, George McMechan, and Gladys Gonzalez. Solutions of multipathing in incident angle, depth, and image time domain. In *SEG Technical Program Expanded Abstracts 2015*, pages 4339–4344. Society of Exploration Geophysicists, 2015.

[64] Xiaoyu Zhang, Qunshan Zhang, George A McMechan, and Gladys Gonzalez. Multipathing in three-parameter common-image gathers from reverse-time migration. *Geophysical Prospecting*, 65(3):669–686, 2017.

[65] XZ Zhang, MO Ostadhassan, AUB Buriti, and A Guedes Barros. The separation of multipath angle domain common image gathers for complex structures. In *3rd Latin American Geosciences Student Conference*, 2015.

[66] Feiyun Zhu, Jun Guo, Zheng Xu, Peng Liao, and Junzhou Huang. Group-driven reinforcement learning for personalized mhealth intervention. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2018.

[67] Xinliang Zhu, Jiawen Yao, and Junzhou Huang. Deep convolutional neural network for survival analysis with pathological images. In *Bioinformatics and Biomedicine (BIBM), 2016 IEEE International Conference on*, pages 544–547. IEEE, 2016.

[68] Xinliang Zhu, Jiawen Yao, Feiyun Zhu, and Junzhou Huang. Wsisa: Making survival prediction from whole slide pathology images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

[69] Andrew Zisserman, Joao Carreira, Karen Simonyan, Will Kay, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, and Mustafa Suleyman. The kinetics human action video dataset. 2017.

BIOGRAPHICAL STATEMENT

Xiaoyu Zhang received his Bachelor of Technology in Geoexploration Technology and Engineering from Jilin University, Changchun, China and Master's degree on Exploration Seismology from UT-Dallas in 2014. After graduation, he worked for a couple of oil and gas companies on data processing and imaging. He come back to pursue his Master degree in Computer Science at The University of Texas at Arlington in Fall 2017 and joined Dr.Junzhou Huangs lab for his thesis. His interests include the theory and various applications of machine learning(Deep Learning), such as the intersection of computer vision and deep learning.