SEMI-SUPERVISED DEEP LEARNING WITH APPLICATIONS IN SURGICAL VIDEO ANALYSIS AND BIOINFORMATICS

by

SHENG WANG

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
May 2020

To my dear family, for all their endless trust, continuous support, and unconditional love.

ACKNOWLEDGEMENTS

There were many people who helped me during my years of Ph.D. study, and I would like to take this opportunity to thank them.

I would like to thank my supervising professor, Dr. Junzhou Huang, for continually 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. Chengkai Li, Dr. Dajiang Zhu, Dr. Jia Rao, for their interest in my research and 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.

My research and coding skills also got polished from my internships in industry. My special thanks go to Dr. Diana Delibaltov, Dr. Dong Wang, and Dr. Zhusong Li. I have learned a lot from them through awesome collaborations. Without the improvement of my general problem-solving ability, some of the chapters in this thesis would not have been possible.

I want to thank all my colleagues from the Scalable Modeling and Imaging and Learning Lab (SMILE), the Computer Science and Engineering Department of the University of Texas at Arlington. 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 UT Arlington.

My special thanks go to my family in China. 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 life.

Finally, I hope the world could win the fight against Covid-19 soon.

April 24, 2020

ABSTRACT

SEMI-SUPERVISED DEEP LEARNING WITH APPLICATIONS IN SURGICAL VIDEO ANALYSIS AND BIOINFORMATICS

SHENG WANG, Ph.D.

The University of Texas at Arlington, 2020

Supervising Professor: Junzhou Huang

In the current era of big data, deep learning has been the state-of-the-art model for various applications. Image-based applications such as image classification, object detection, image segmentation, benefit most from deep learning networks. One reason for the successful applications of deep learning is that there are a large number of labeled training samples for the model to learn from. People are interested in reducing the cost of getting labeled training samples, and there are various research going on with unsupervised, semi-supervised, and self-supervised deep learning. The cost of health-related data is even higher. Labeling the surgical videos with tools being used and surgical phase needs surgical related domain knowledge, it is not feasible to use general cloud labeling. Getting molecule properties even cost more since it usually needs expensive laboratory experiments. How to utilize the unlabeled data to improve the model performance attracts increasing research interests. In this thesis, we aim at proposing semi-supervised deep learning models to introduce unlabeled data into model training to get better model performance. Specifically, this thesis focuses on developing semi-supervised deep models for in surgical tool presence detection problem, and molecular property prediction problem.

Surgical tool presence detection is one of the key problems in automatic surgical video content analysis. Solving this problem benefits many applications, such as the evaluation of surgical instrument usage and automatic surgical report generation. Given the fact that each video is only sparsely labeled at the frame level, meaning that only a small portion of video frames will be properly labeled, existing approaches only model this problem as an image (frame) classification problem without considering temporal information in surgical videos. In this thesis, we discuss from a supervised deep neural network to a semi-supervised frame, which utilizes the information from both labeled and unlabeled frames to solve this problem with different components to capture the spatial and temporal information of surgical videos.

With the rapid progress of AI in both academia and industry, Deep Learning has been widely introduced into various areas in drug discovery to accelerate its pace and cut R&D costs. Among all the problems in drug discovery, molecular property prediction has been one of the most important problems. Unlike general Deep Learning applications, the scale of labeled data is limited in molecular property prediction. To better solve this problem, Deep Learning methods have started focusing on how to utilize tremendous unlabeled data to improve the prediction performance on small-scale labeled data. In this thesis, we discuss a semi-supervised model named SMILES-BERT, which consists of the attention mechanism based Transformer Layer. A large-scale unlabeled data has been used to pre-train the model through a masked SMILES recovery task. Then the pre-trained model could easily be generalized into different molecular property prediction tasks via fine-tuning.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure Page

LIST OF TABLES

Table Page

CHAPTER 1

INTRODUCTION

This thesis focuses on developing semi-supervised deep learning algorithms and models in surgical video analysis and bioinformatics, including tasks as surgical tool presence detection and molecule property prediction.

1.1 Motivation

In the current era of big data, deep learning has been the state-of-the-art model for various applications. Image-based applications such as image classification, object detection, image segmentation, benefit most from deep learning networks. One reason for the successful applications of deep learning is that there are a large number of labeled training samples for the model to learn from. People are interested in reducing the cost of getting labeled training samples, and there are various research going on with unsupervised, semi-supervised, and self-supervised deep learning. The cost of health-related data is even higher. Labeling the surgical videos with tools being used and surgical phase needs surgical related domain knowledge, it is not feasible to use general cloud labeling. Getting molecule properties even cost more since it usually needs expensive laboratory experiments. How to utilize the unlabeled data to improve the model performance attracts increasing research interests. In this thesis, we aim at proposing semi-supervised deep learning models to introduce unlabeled data into model training to get better model performance. Specifically, this thesis focuses on developing semi-supervised deep models for in surgical tool presence detection problem, and molecular property prediction problem.

First of all, this thesis focuses the sparsely-labeled surgical videos to solve the surgical tool presence detection problem. In the past decades, the operating room (OR) has experienced a series of significant transformations, evolving into a highly complex and technologically rich environment [1, 2]. Among all the transformations, computer-assisted intervention (CAI) systems play an increasingly vital role in current surgical performance [3]. To build context-aware CAI systems, a lot of researchers have been working on various computer vision-related tasks, such as surgical tool detection [4, 5] and tracking [6, 7, 8], surgical activity recognition [9], and surgical phase recognition [10, 4]. Among all the tasks for CAI systems in minimally invasive surgery (MIS), surgical tool presence detection is one fundamental and significant task to be solved. Surgical tool presence detection [11, 12] is to automatically detect what surgical tools are being used at a specific time during surgery. Understanding what tools are being used is the basis of surgical tool localization [4], tracking [6], as well as robot-assisted surgery [13]. It would also benefit the surgical phase recognition task since there is a high correlation between the surgical phase and tool usage [10]. With the help of surgical tool presence detection, CAI systems could generate a real-time warning to the surgeons if any abnormal tool usage is realized during a surgery [14]. Besides, surgical tool presence detection could facilitate the surgeon training, review, and skill assessment [15, 16]. The surgical tool presence detection problem is different from surgical tool localization [17] or general object detection [18, 12] that it requires the awareness of the presence of surgical tools instead of their locations. However, this task still challenging due to three reasons. First, the endoscopic camera in MIS restricts the field-of-view (FoV), making detecting tools more difficult [1]. Second, multiple surgical tools could be used at the same time and tools could have partial presence and occlusion, which makes it even harder to detect; Third, the datasets could be very imbalanced since the frequencies of different surgical tools being used vary a lot [19]. Besides, the sparsely labeled video is structured by continuous frames in temporal dimension, with only a small portion of the frames having label. It is challenging to utilize information from both labeled and unlabeled frames, as well as both spatial and temporal information.

Second, this thesis investigates the problem of molecular property prediction. The capability of accurate prediction of molecular properties is an important key in the chemical and pharmaceutical industries. It benefits various academic areas and industrial applications such as improvement to rational chemical design, reducing R&D cost, decreasing the failure rate in potential drug screening trials, as well as speeding the process of new drug discovery [20]. The key problem of introducing Deep Learning into this area lies on embedding graph-like molecules onto a continuous vector space. Then the representations could be used for various application such as molecular properties classification, regression, or new generating new molecules. Molecular fingerprints are the names of molecular representation. Instead of computing a basic property, Molecular fingerprints provide a description of a specific part of the molecular structure [21]. However, traditional molecular fingerprints require intensive manual feature engineering and strong domain knowledge. Besides, this kind of fingerprints is highly task-dependent, not general enough for other property prediction tasks. [22] The current success of deep learning in various areas and applications, e.g., image classification [23, 24, 25], video understanding [26, 27], medical imaging [12, 28, 29], bioinformatics [30, 31, 32, 33] and other applications [34, 35, 36, 37, 38] demonstrates that deep learning is a powerful tool in learning feature from data and giving a task-related prediction. An increasing number of publications have introduced deep learning into molecular fingerprint learning [30, 32, 39, 40]. The success of the current deep learning methods highly relies on a large-scale labeled training dataset. For many areas, the labeled sample number of image classification or natural language translation could easily reach several million or even more. However, it is not the same situation with molecular property prediction. The cost of obtaining such scale of molecular properties with screening experiments is exceptionally high. It is similar to the case in natural language modeling that they have almost unlimited unlabeled data while a tiny portion has a label. The state-of-the-art framework to utilize the unlabeled data is the pre-training and fine-tuning framework [41]. It pre-trains the model in an unsupervised fashion then fine-tune the model on labeled data. Seq3seq Fingerprint model [32] first starts using this framework to involve unlabeled data in model training to improve the prediction performance. However, Seq3seq model uses an encoder-decoder structure, and the decoder is used as a scaffold and does not contribute to the final prediction. The motivations of our paper are two-folded. First, we would like to build a powerful model utilizing the essential information in unlimited unsupervised learning. The model used for pre-training will all take part in used in the fine-tuning stage. Second, we would like our model to naturally support parallel training to reduce pre-training time.

1.2 Our Techniques

For the surgical tool presence detection problem, we first describe our model for the M2cai challenge, a deep supervised image classification solution to simplify the structured data to images. Then we discuss introducing the unlabeled data from surgical videos into training. To utilize the temporal information of the surgical videos for detection, it is not easy to apply current methods straightforwardly. Since almost all current surgical tool detection datasets are sparsely labeled at the frame level, using fixed length frames around the labeled image as a video could either introduce noise or lack enough temporal information. It might not offer enough temporal information when the video length is too small, while it might introduce noise when the video length is too large. Besides, if we use continuous frames around the labeled image as a video, the length of videos in this problem will not be long enough or the variation of the frame contents will not be large enough to learn long-range temporal dependency with Recurrent Neural Networks (RNNs) such as Long Short-Term Memory (LSTM) [42, 32, 43] for general video understanding. We propose a novel deep neural network model named Surgical Tool Graph Convolutional Networks (STGCN) combining the power of both Convolutional Neural Networks (CNNs) and Graph Convolutional Networks (GCNs) [44]. We model the problem as a video classification problem by using the sparsely labeled frame and the neighbor frames around it. STGCN uses DenseNet [23] as our backbone to learn the spatial features from the input images and extracts the features directly from the videos with inflated 3D DenseNet. Then it applies GCNs along the temporal dimension to learn better feature with consideration of the relationships among continuous frames. Based on this, we also propose a general semi-supervised training framework consisting of an spatial encoder and a temporal encoder which could adapt different deep neural network as instantiations of the framework. From STGCN, we propose a generalized semi-supervised framework consisting of a spatial encoder and a temporal encoder to solve this problem. Each of the spatial encoder and temporal could be instantiated with different state-of-the-art deep models or components.

For the molecular property prediction problem, firstly, we review the semi-supervised seq3seq fingerprint [32]. The backbone of the seq3seq network is the seq2seq [43, 45] model. Seq3seq uses a semi-supervised fashion to train from both unlabeled and labeled molecular SMILES sequences. The training is done with two tasks: a self-recovery task and a property prediction task. Then we discuss our proposed pre-training and fine-tuning two-stage framework named SMILEBERT based on the natural language modeling work BERT [46]. The neural network structure is a fully convolutional stack of Transformer layers. In the pre-training task, SMILEBERT is trained with unsupervised learning mechanism Masked SMILEs Recovery on large scale unlabeled data. In the Masked SMILEs Recovery task, the input SMILEs will be randomly masked/corrupted, and the model is being trained to recover the original SMILEs according to the information lying in the unmasked part of the input. After that, the model only needs a slightly fine-tuning with the labeled dataset to have excellent prediction performance. The proposed SMILEBERT contains several benefits than the existing methods: 1) different from Seq2seq or Seq3seq model, SMILEs-BERT does not require an encoder-decoder structure which is more efficient and the model could be more complicated given the same GPU memory; 2) SMILEBERT is more natural to parallel training because of the fully convolutional structure; 3) The random masking method will having SMILEBERT more general and able to avoid overfitting; 4) The attention mechanism is used in the Transformer layer which could potentially improve the prediction performance.

1.3 Thesis Overview

Finally, we provide the overview of this thesis in brief. In Chapter 2, we present the series of our deep learning approaches from supervised to semi-supervised deep models to handle the surgical tool presence detection problem. Then, Chapter 3 focuses on our model for molecular property prediction problem with large scale unsupervised pre-training. In Chapter 4, a conclusion of the thesis is given.

CHAPTER 2

Deep Learning and Graph Deep Learning in Surgical Tool Presence Detection

2.1 Introduction

Automatic content analysis of surgical videos recorded by an endoscopic camera in minimally invasive surgery is significant for many functions in the operating room of the future [2], such as analysis of the operation steps, review of the techniques employed, evaluation of instrument usage, and automatic surgical report generation [47]. Among all the tasks of surgical video content analysis, one crucial problem is surgical tool presence detection, to detect which surgical tools are being used at a certain time during surgery. The problem is different from surgical tool detection [17] or object detection [18, 12] since it does not require the awareness of the location of surgical tools or general objects. However, the problem is challenging due to several reasons: First, multiple surgical tools could be used at the same time. Second, different tools could have partial presence and occlusion which makes it even harder to detect. Third, since the frequencies of different surgical tools being used vary a lot, the data could be very imbalanced among certain surgical tools [4].

Existing approaches and models solve this problem by engaging multi-label image classification: sampling every frame with ground truth as an image dataset, learning features from each still image and then perform classification [4, 17, 48, 49, 19, 50]. There are two ways of feature extraction. One is to use manually hand-crafted features or pre-designed features, e.g., SIFT features. The other is to use deep neural networks such as convolution neural networks (CNNs) to extract high-level features. After applying deep neural networks, the classification accuracy generally improves.

However, one key piece that is still missing from the current methods is the information along the temporal dimension, which is the nature of videos. As shown in Figure 2.1, almost all surgical tool detection datasets are labeled sparsely, i.e. the tools being used are not labeled for every frame. Only a very **tiny portion** (usually only a few percentages) of video frames are manually labeled. The insufficient label information leads to a huge challenge for the research of machine learning based surgical tool presence detection. To address this problem intuitively, the temporal information from neighbor frames could help the presence detection and should provide better performance than utilizing only the labeled image. For instance, one tool might be occluded at a certain frame and it can be very difficult to recognize it from the complex background by one single image. However, when using a continuous sequence of frames, even slight movement of the surgical tool could be noticed and help the tool get detected correctly.

To utilize the temporal information of the surgical videos for detection, it is not easy to apply current methods straightforwardly. Since almost all current surgical tool detection datasets are sparsely labeled at the frame level, using fixed length frames around the labeled image as a video could either introduce noise or lack enough temporal information. It might not offer enough temporal information when the video length is too small, while it might introduce noise when the video length is too large. Besides, if we use continuous frames around the labeled image as a video, the length of videos in this problem will not be long enough or the variation of the frame contents will not be large enough to learn long-range temporal dependency with Recurrent Neural Networks (RNNs) such as Long Short-Term Memory (LSTM) [42, 32, 43] for general video understanding.

In this thesis, we propose three models: 1) A deep ensemble network combining the power of GoogleNet and VGGNet to detect surgical tool presence on the labeled image level. The proposed model won the M2CAI surgical tool presence detecion challenge in 2016. 2) A novel deep neural network model named Surgical Tool Graph Convolutional Networks (STGCN) combining the power of both Convolutional Neural Networks (CNNs) and Graph Convolutional Networks (GCNs) [44]. We model the problem as a video classification problem by using the sparsely labeled frame and the neighbor frames around it. STGCN uses DenseNet [23] as our backbone to learn the spatial features from the input images and extracts the features directly from the videos with inflated 3D DenseNet. Then it applies GCNs along the temporal dimension to learn better feature with consideration of the relationships among continuous frames. 3) A novel semi-supervised learning framework for surgical tool presence detection. The proposed framework consists of two encoders: a spatial encoder and a temporal encoder. The spatial encoder extracts spatial features from the unlabeled and labeled frames independently. The temporal encoder uses the temporal attention mechanism that encodes the spatial and temporal features together for the final detection. Other than solving the surgical tool presence detection problem as an image classification problem, we model the problem as a video classification problem, and the proposed framework is trained on the video segments containing both sparsely labeled frames and their neighbor unlabeled frames.

To fully demonstrate the superiority of our model, we compare our image model with other participants in M2CAI surgical tool challenge. Then we compare our video models with state-of-the-art methods at the publication time on two most recently developed datasets: M2cai-tool and Cholec80 [4].

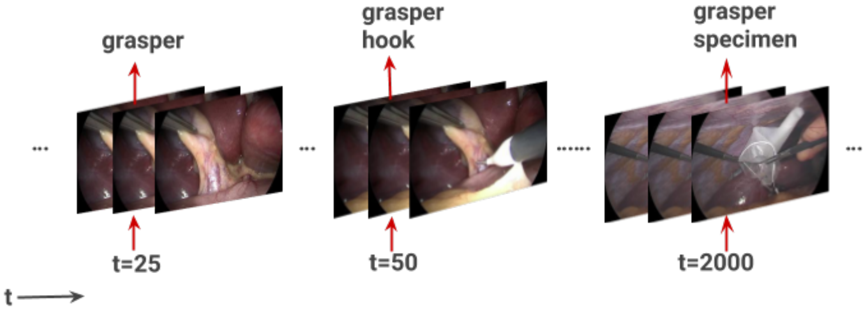


Figure 2.1. Sparsely labeled surgical tool detection dataset. In this dataset, the tools being used in one image is labeled every 25 frames. Existing methods only use the labeled images for model training. In this paper, we propose to use both the labeled frame and the unlabeled frames around it as a video for model training..

2.2 Related Work

2.2.1 Surgical Tool Detection

Early methods for surgical tool detection focused on extracting low-level manually designed features including color features, gradient features, shape features, texture features, and combinations of these features [51, 52, 14, 1, 53, 54, 55, 56]. Color features were popular in surgical tool detection, and the surgical images could be represented in different color spaces such as RGB, HSV, Cie XYZ spaces [14, 56]. However, color features were not robust to visual ambiguities caused by shadows and lighting. Gradient features were less used in surgical tool detection as well [14, 53, 54] since they were good at describing oriented edges and corners but suffer heavily from noise which is common in medical images. Since the surgical tools are known before surgeries, the shape features of the surgical tools could be extracted for the detection [52, 55]. The texture features like SIFT, SURF, Color-SIFT were widespread since these features were more robust than the gradient features [51, 57, 58, 54]. Combinations of these features were studied in different works [52, 14, 1] to improve the tool detection performance. Many early methods also relied on a set of assumptions, or prior knowledge of MIS [51, 59, 60, 17]. Such prior knowledge includes the tool shape constraints, tool location constraints. Though early methods combined the power of prior knowledge and combinations of different low-level features, the low-level features were not robust and did not provide strong representations for the detection problem.

The current success of deep learning in various areas and applications, e.g., image and video understanding, medical imaging, and bioinformatics, demonstrates that deep learning is a powerful tool in learning features from data and good at task-related prediction [61, 62]. There is an increasing number of deep learning models being proposed to improve the surgical tool detection performance [48, 49, 50, 10, 63, 4, 19], and the overall performance has been largely improved. EndoNet [4] first proposed to use CNNs to train a tool detection model on labeled images. Along with EndoNet, a larget MIS video dataset named Cholec80 has been released for researchers to contribute better models and solutions. Part of the datasets has been used for the M2CAI tool presence detection challenge. The winner of M2CAI tool detection challenge, [19], modeled the surgical tool detection problem as a multi-label image classification problem and trained a VGGNet and an InceptionNet for tool detection. The authors ensembled the results of these two deep models as the final detection. After that, ZIBNet was proposed by [63] to handle the data imbalance problem in M2cai-tool dataset by data augmentation. Since the surgical tool presence problem is slightly simpler than surgical tool localization, two methods have been proposed to further improve the detection performance by labeling extra localization information of surgical tools to the original dataset [48, 50]. [49] proposed a coarse-to-fine model named AGNet, cascading of two components: the first component is an attention model as a global network to detect the areas with high possibilities to contain the surgical tools and the second component is a local model to detect the tools from selected areas with high possibilities. Compared to all the models focusing on the surgical tool detection problem, AGNet has the best performance on M2cai-tool dataset.

According to the regulation of surgery procedures, surgeons perform specified operations with corresponding surgical tools for different surgery phases. Thus, there is a high correlation between surgical tool usages and surgical phases. EndoNet [4] and MTRCNet-CL [10] proposed to solve the two problem with multi-task learning. EndoNet included the surgical phase as an extra feature for the surgical tool detection. MTRCNet-CL proposed an end-to-end CNN-LSTM model with a correlation loss to learn from both tasks. Multi-task learning with the two tasks improves each task's performance. However, these models require the datasets to have surgical tool labels as well as the surgical phase labels.

2.2.1.1 Graph Convolutional Networks

Until recent years, very little attention has been devoted to the generalization of neural network models to more general structure such as graphs or networks [64, 65]. The deep models handling the graph-like structure are named Graph Convolutional Networks (GCNs).

Our work is motivated by recent work on human recognition [66] using GCN as one crucial part of their proposed deep neural network model. In this work, the authors built a graph containing nodes corresponding to different object proposals aggregated over video frames. Different from this work, we model the feature extracted from each frame as a node and build the graph as the relationship within the continuous frames of a video segment to learn better feature with temporal information.

2.2.2 Semi-Supervised Learning

The success of deep learning methods relies on a large scale labeled dataset. While in some applications such as medical imaging and bioinformatics, the labeled data is harder and more expensive to get. There have been many researchers working on semi-supervised learning [67]. Semi-supervised learning aims to use relatively easy-to-get unlabeled data to improve model performance when the number of labeled data is limited. There are several successful attempts of semi-supervised learning on medical imaging and surgical videos [68, 69, 70]. As shown in Figure 2.1, the sparsely labeled surgical videos contain more unlabeled frames, which could have a high correlation to the labeled frames. It is reasonable to assume that semi-supervised learning combining these unlabeled frames would be beneficial to the surgical tool detection problem.

2.2.3 Video Understanding and Temporal Attention

Meanwhile, many researchers focus on video inference for the better ability of computer video understanding. A considerable number of cutting edge approaches have been proposed to improve the video understanding performance, and several large-scale datasets have been built to promote related research [27, 71, 72, 73]. One challenging problem in video understanding is how to utilize temporal information from videos. Either recurrent Neural Networks (RNN) and optical flows related methods are good at capture the temporal features. In surgical videos, there has also been some surgical video understanding work on the surgical phase recognition with Long short-term memory (LSTM) [74, 10, 70]. Different from the surgical tool presence detection problem, surgical phase recognition demands to model long term temporal information on a whole surgical video. The frames feeding into LSTMs are very visually different. However, in semi-supervised surgical tool detection, the continuous frames are visually similar. Thus, RNNs based methods might not serve as a good fit in our problem. We need the temporal attention technique to capture the slightest difference between the continuous frames.

Until recent years, very little attention has been devoted to the generalization of neural network models to more general structures such as graphs or networks [64, 65]. The deep models handling the graph-like structure are named Graph Convolutional Networks (GCNs). Our ST-GCN model is motivated by recent work on human recognition [66] using GCN as one crucial part of their proposed deep neural network model. In this work, the authors built a graph containing nodes corresponding to different object proposals aggregated over video frames. Different from this work, ST-GCN models the spatial feature extracted from each frame as a node and build a similarity graph as the relationship within the continuous frames of a video segment to learn better feature with temporal information. In the meantime, self-attention based methods [46, 26] has been widely used in modeling the temporal relationship in natural language modeling. The temporal encoder in the proposed ST-TAN is motivated by such models that we introduce the self-attention module from language modeling into the surgical tool detection problem.

2.3 Methodology One: Image Classification

Our method follows two main steps: training the CNN models – VGGNet and GoogLeNet, then using model ensembling to combine the results of the models to get the final results. Before giving the details of the two steps, we will describe the surgical tool presence detection as a multi-label classification problem.

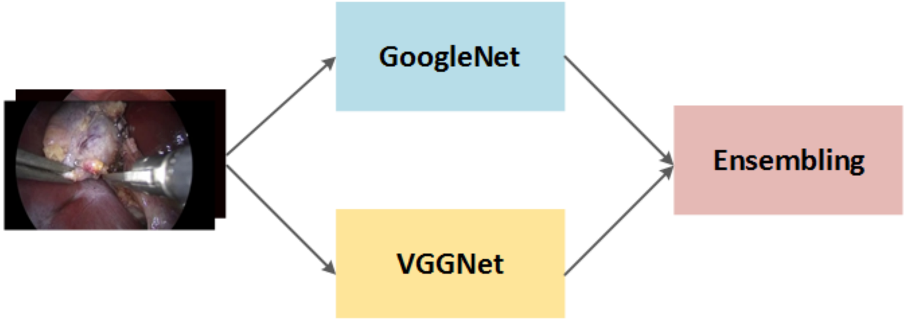


Figure 2.2. Pipeline for our tool presence detection method. The left side shows the training image samples, the middle shows two deep neural networks trained from the training images, the right is the ensemble learning technique combining the results of the two models..

2.3.1 Multi-label Classification

Traditional multi-class classification is the problem of classifying instances into one of the more than two classes, and each instance belongs to only one class. Different from multi-class classification, multi-label classification allows each instance to belong to one or more than one classes. Multi-label classification is a generalization to multi-class classification. In real-world problems, multi-label classification tasks are ubiquitous. For instance, in text categorization, each document can belong to more than one predefined topics, such as sport and health.

The surgical tool presence detection problem can also be viewed as a multi-label classification problem. It is because that each image which we extract as image frames from the surgery videos may contain one or more than one surgical tools. Thus, each image can belong to one or more than one classes. In this way, we can use multi-label classification methods for surgical tool presence detection. The two common methods for multi-label classification are problem transformation and algorithm adaption. Problem transformation decomposes the multi-label classification problem into multiple independent binary classification problems. Algorithm adaptation methods [75] design or adapt algorithms to solve multi-label classification directly. In the proposed method, we use problem transformation method to convert the multi-label classification problem into several independent binary classification problems. Each of the binary classifiers is to detect if one kind of the tools is used in the images.

2.3.2 VGGNet and GoogLeNet

VGGNet [76]. VGGNet is a deep CNN architecture with 16 layers. Different from other deep CNN architectures, the convolutional layers in VGGNet use very small (3 × 3) convolution filters. In our training process, we initialize the network weights with the method mentioned in [77]. Rectified Linear units (ReLU) [78] is used as the activation function VGGNet. The batch size used in VGGNet is 32.

GoogLeNet [79]. GoogLeNet is a deep convolutional neural network architecture with 22 layers. GoogLeNet integrates several inception modules inside. The inception modules can increase the depth and width of the network while keeping the computational complexity. GoogLeNet has six more layers than VGGNet but three times fewer parameters compared with VGGNet. GoogLeNet has the ability for multi-scale processing and has achieved state of the art for classification and detection in the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC14). In our training process, we use Leak ReLU [80] as the activation function. The batch size used in GoogLeNet is 64.

For both VGGNet and GoogLeNet, we use sigmoid cross-entropy as the loss function and use batch normalization after convolutional layers.

2.3.3 Model Ensembling

An ensemble [81] consists of a set of independently trained classifiers whose predictions are combined as the final prediction when classifying new instances. Many

Index	T1	T2	T3	T4	T5	T6	T7
Number	10967	635	14130	411	878	953	1504

Table 2.1. The numbers of training images for each surgical tool.

research studies have shown that good combination of the predictions of multiple classifiers can produce a better classifier.

We use ensembling in our methods for the three following reasons: First, according to the theory of ensemble learning, it is promising to get better classification performance from the ensemble of individually trained classifiers. Second, the process of training deep neural networks tends to overfit the training dataset even if some techniques for avoiding overfitting such as early stopping and Dropout are used in training process or network architecture. Third, data sets provided by challenges always have a larger variance. Thus, even if we get good performance on the validation data set, we cannot assure it will have similar performance on the testing data set.

In the proposed method, we use model averaging to ensemble the predictions from all trained GoogLeNet and VGGNet together to get the final prediction. Simply speaking, we have a prediction probability for each image from each of the trained models and we calculate the average of the probabilities as the final probability for the image.

2.4 Experiments

To evaluate our method, we have submitted the results of our method to the M2CAI surgical tool presence detection challenge[1] and add some experimental analysis by using the ground truth of the challenge testing data set.

[1] M2CAI Surgical Tool Presence Detection Challenge 2016: http://camma.u-strasbg.fr/m2cai2016/

2.4.1 Data Description and Augmentation

This dataset from M2CAI surgical tool presence detection contains 15 videos of laparoscopic cholecystectomy procedures from University Hospital of Strasbourg / IRCAD (Strasbourg, France). The dataset is split into two parts: the training subset (containing ten videos) and the testing subset (5 videos) by the challenge organizers. In the 15 videos, there are seven kinds of surgical tools in total as shown in Figure 2.4: grasper, hook, clipper, bipolar, irrigator, scissors and specimen bag. We notate the seven tools from T1 to T7 for short.

Table 2.1 shows the number of training images for each kind of surgical tools in the training set. From the table we can find that the dataset is imbalanced, which makes it more difficult for the models to handle.

2.4.2 Data Preprocessing and Augmentation

Data Preprocessing. We extract the images which have ground truth labels from the ten training videos and resize them into the same size (224 × 224) since the videos have different dimensions. We use the data from the ten training videos as training and validation sets. For the five testing videos, we extract the images as required by the challenge as the testing set. We also resize them into 224 × 224.

Data Augmentation. We introduce three kinds of data augmentation methods: horizontal flipping, vertical flipping, and rotation. In the implementation, we do not generate the augmented data set before training. Instead, we dynamically augment each image via each of the three augmentation methods in each epoch of the training process. For each image in a certain training epoch, it has 0.5 probability to be horizontal flipped. It also has 0.5 probability for other two augmentations. The three augmentation methods are taken independently. Thus, we augment our training data set in a dynamic way to better train the models. We do not augment our validation set or testing set.

2.4.3 Experiment Settings

In the training stage, for both VGGNet and GoogLeNet, we use the training set given by the challenge. We randomly choose 90% data as a training set, and 10% as the testing set five times. Thus, we train ten models in total with five VGGNet models and five GoogLeNet models. We train ten models to let different models have different training data. Then after averaging the ten models, the ensemble will hardly overfit the training data.

2.4.4 Experimental Results

In this paper, we use the same evaluation protocol used in the challenge. We use the final prediction ensemble from the ten models on the testing set as the final submission to M2CAI surgical tool presence detection challenge. The mean accuracy precision (mAP) values of all the participants are listed in Table 2.2. The proposed methods have better mAP than the other methods. The method by Sahu et al. has the second best performance by introducing the temporal information to help classification. It demonstrates that our model has excellent performance even not considering the temporal information. Table 2.3 shows the mAPs for each kind of the surgical tools. Our method is still affected by the imbalance of the data set. Further effort should be taken into handling data imbalance.

Methods	Mean AP
Proposed	**63.8**
Sahu et al.	61.5
Twinanda et al. [4]	52.5
Zia et al.	37.8
Luo et al.	27.9
Letouzey et al.	21.1

Table 2.2. The leader board of M2CAI surgical tool detection challenge. The evaluation metric is mean accuracy precision.

Index	T1	T2	T3	T4	T5	T6	T7
mean AP	81.4	62.8	88.2	49.8	49.8	35.3	55.2

Table 2.3. The mean AP values for each of the seven tools evaluated. These values are computed from our final submission

2.5 Methodology Two: Graph Convolution Based Video Classification

2.5.1 Problem Definition

2.5.1.1 Image classification.

Existing methods for surgical tool detection models the problem as an multi-label image classification problem. Given the image x_t at frame t, models are trained to get the prediction for the input image $F(x_t)$ close to its groundtruth y_t.

2.5.1.2 Video classification.

In this paper, we propose to use not only the labeled image but also the neighbor images as a video segment for model training and evaluation. Thus, the problem becomes that given a video segment corresponding to the t frame $[x_{t-l}, ..., x_t, ..., x_{t+l}]$, where l is the number of frames before and after the labeled frame image we take into consideration, models are trained to get the prediction for the input video $F(x_{t-l}, ..., x_t, ..., x_{t+l})$ close to its groundtruth y_t.

2.5.2 Model Overview

As shown in Figure 2.3, the proposed STGCN contains several components. To get the features from the input video, we use an inflated 3D DenseNet-121 [71, 23] to get the representation of each frame in the video. We take the representation of each frame as a node and build a similarity graph on these nodes. By applying GCNs on the constructed graph, the GCNs will adaptively generate the features considering the relationships among the nodes in the graph, i.e., the temporal relationship in continuous frames. After that, we use pooling over all the nodes corresponding to continuous frames. We note the pooling layer as temporal pooling since what it does is applying the pooling on the temporal dimension. The details of each component will be discussed in the following sections.

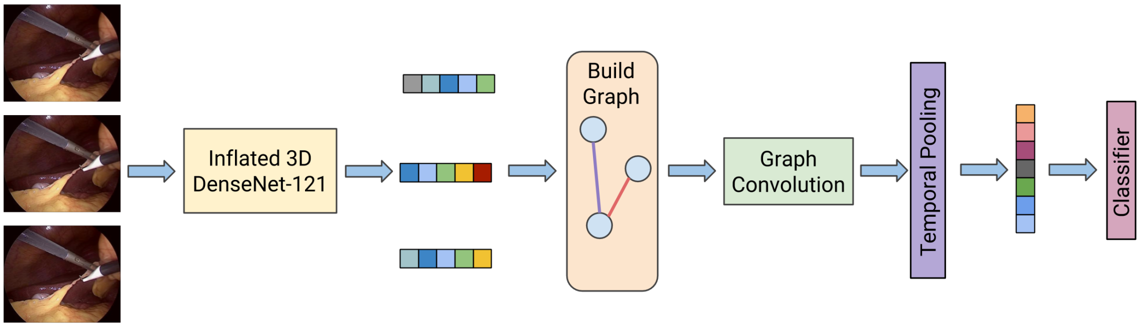


Figure 2.3. The overview of the proposed STGCN..

2.5.3 Inflated 3D DenseNet

Different from most deep convolutional neural networks, DenseNet [23] connects all the convolutional layers in pairs when their spatial output sizes are the same. The output of each feature maps also serves as the input of all following convolutional layers. The idea is similar to Residual Networks. However, it can reuse all the features in the network. This sort of network almost exhaustively maximizes the network capacity to squeeze its spatial feature extraction and prediction power. Also, the network can alleviate the vanishing-gradient problem, strengthen feature propagation and substantially reduce the number of the parameters in the network.

In our proposed model, we use DenseNet to learn and extract spatial features for each frame in the input video. To adapt DenseNet for video input, the original DenseNet needs to be inflated to 3D ConvNet (I3D) [71, 23]. That is, to support the input video of length t, a 3D kernel with $t \times k \times k$ dimensions can be inflated from a 2D $k \times k$ kernel by copying the weight t times and rescaling by $1/t$. In our implementation, we use 11 as the number of frames. The growth rate is 32 as the default number for DenseNet-121.

2.5.4 Graph Convolutional Networks

We apply GCNs [64] in the proposed framework to better capture the temporal relationship along the continuous frames.

2.5.4.1 Similarity Graph Building.

For a video input $X = [x_{t-l}, ..., x_t, ..., x_{t+l}]$ with length N, where x_t is with the dimension of d, containing the labeled surgical tools while others not. We use the output of the fully-connected layer right after the fourth dense block from our inflated DenseNet-121 model to get the feature representations noted as

$$[f(x_{t-l}), ..., f(x_{t-1}), f(x_t), f(x_{t+1}), ..., f(x_{t+l})]$$

. We regard the representation for each frame as one vertex (node) v_k of a graph, and use the similarity S_{ij} between each pair of nodes (v_i, v_j) as the corresponding edge of the graph. Thus, the graph could reflect the temporal relationship of the continuous frames.

There are quite a few different methods to build the similarity graph. In the proposed STGCN, we use the **cosine similarity** to build the graph as

$$S_{ij} = \frac{f(x_i) \cdot f(x_j)}{\|f(x_i)\| \, \|f(x_j)\|}, \tag{2.1}$$

and we can get the similarity graph G after normalizing each row of S as

$$G_{ij} = \frac{e^{S_{ij}}}{\Sigma_{j=1}^{N} e^{S_{ij}}}. \tag{2.2}$$

2.5.4.2 Graph Convolutional Layer.

After building the similarity graph, the graph convolutional layer could be represented as

$$Z = GXW, \tag{2.3}$$

where W is the weight mapping feature of each node to another dimension. The graph convolutional layer could not only map the feature as a general convolutional layer, but also take the graph information (temporal relationship among the frames in the input video) into consideration. In the surgical tool detection problem, graph convolutional layer could learn features while adaptively reference the relationship among the frames to generate the correct prediction.

The graph convolutional layers could be stacked as a deep GCNs or in general CNNs by

$$X^{(l)} = GX^{(l-1)}W^{(l-1)}, \tag{2.4}$$

where $X^{(l-1)}$ is the feature map as the input to current graph convolutional layer, $W^{(l-1)}$ is the weight. $X^{(l)}$ is the output of current layer as well as the input of next layer.

In our proposed model, we use a residual variation of the graph convolutional layer as

$$X^{(l)} = \sigma\left(GX^{(l-1)}W^{(l-1)}\right) + X^{(l-1)}, \tag{2.5}$$

where $\sigma(\cdot)$ is the activation function after the graph convolutional layer and we add $X^{(l-1)}$ to the output of the layer as a residual component.

2.5.5 Temporal pooling

The feature after the last graph convolutional layer contains N features for the N frames. Then we add a temporal pooling layer to combine all the N features from N frames in the video. Temporal pooling layer has no difference than general pooling layer that it aggregates the features along the temporal dimension. It should not be a crucial factor in the performance of the proposed model since the features for the pooling layer has utilized the temporal information with GCNs. However, we still try different pooling strategies in STGCN to seek potential improvement. There are a lot of methods for pooling such as l_p pooling, average pooling, max pooling, and max-min pooling [82]. In later ablation experiments, we will show the performance of different pooling methods on Cholec80 dataset.

Given a sequence of N d-dimensional dense features after GCNs as $x^{(i)}$, where i is from 1 to N, temporal pooling pools the features along the time dimension.

Assume the N-dimensional feature after temporal pooling as $\tilde{x}$, for **max temporal pooling**, $\tilde{x}_k = max(x_k^{(i)})$ where i from 1 to N, for **average temporal pooling**, $\tilde{x}_k = \frac{1}{N}\sum_{i=1}^{N} x_k^{(i)}$ and for l_p **temporal pooling**, $\tilde{x}_k = \sqrt[p]{\sum_{i=1}^{N}\left(x_k^{(i)}\right)^p}$ where k is from i to d for all temporal pooling methods. For **max-min pooling**, we apply a simple version of max-min pooling, which could be computed as:

$$\tilde{x}_k = max(x_k^{(i)}) + \alpha min(x_k^{(i)}), \tag{2.6}$$

where α is a hyperparameter balancing the weights of max pooling and min pooling.

2.6 Experiments

2.6.1 Implementation Details

2.6.1.1 DenseNet.

We use DenseNet-121 pretrained from ImageNet to continue training on surgical tool detection datasets for a multi-label image classification. Then we inflate the trained DenseNet to 3D DenseNet. To avoid using temporal information in the inflated DenseNet, we keep all the dimension of kernels in either dense blocks or other convolutional/pooling layers as 1. Thus, all the temporal information is used in the GCNs part of the proposed model. We fix the length of the video segment around each labeled image to 11 to train the GCNs and following classifier. The DenseNet is trained with Adam optimizer with learning rate 0.0001 for 200 epochs. The learning rate will be decayed if the training loss does not decrease after three continuous training epochs.

2.6.1.2 GCNs.

After extracting the feature presentation for each frame from Inflated DenseNet-121, we input the features along with the similarity graph into the GCNs. The feature we get from the inflated DenseNet-121 has the dimension of 1024. In our GCNs, we use one graph convolutional layer which maps the input feature from 1024 dimensions to 1024 dimensions. Then the temporal pooling layer is added to pool the features along the temporal dimension. After that is followed by a layer maps 1024 dimensions feature to the number of surgical tools for classification. In GCNs, both batch normalization and dropout are added after the graph convolutional layer. Batch normalization is also added before the graph convolutional layer. We train the GCNs with Adam optimizer with learning rate 0.0001 for 300 epochs. The dropout rate is set as 0.75 in our training. The same learning rate decay strategy is used as the one in training DenseNet. For max-min pooling, we fix the hyperparameter α to 0.75.

2.6.2 Data Description

2.6.2.1 M2cai-tool dataset [4].

This dataset from M2CAI surgical tool presence detection challenge contains 15 videos of laparoscopic cholecystectomy procedures from the University Hospital of Strasbourg/IRCAD (Strasbourg, France). The dataset is split into two parts: the training subset (containing 10 videos) and the testing subset (5 videos) by the challenge organizers. The videos are recorded at 25 fps and labeled at 1 fps (one labeled frame in every 25 frames). There are 23287 training samples and 12541 testing samples. The evaluation process only considers the labeled frames in testing dataset.

In this dataset, there are seven kinds of surgical tools in total as shown in Figure 2.4: grasper, hook, clipper, bipolar, irrigator, scissors, and specimen bag.

2.6.2.2 Cholec80 dataset.

The Cholec80 dataset is larger than M2cai-tool dataset. It contains 40 videos (86304 labeled frames) for training and 40 videos (98194 labeled frames) for testing. The Cholec80 is also from the University Hospital of Strasbourg/IRCAD and has the same recording rate, labeling rate, and tool set as M2cai-tool dataset.

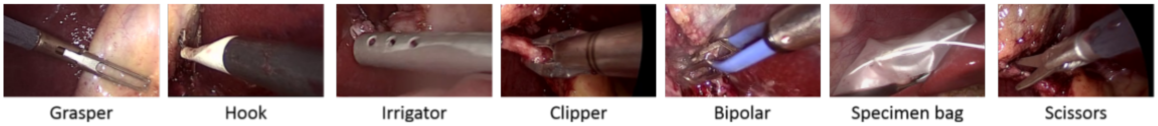


Figure 2.4. The surgical tools used in M2cai-tool and Cholec80 datasets. Both of the datasets have the same seven surgical tools..

2.6.2.3 Validation Sets

For both M2cai-tool and Cholec80 datasets, we split 10% samples from training sets as validation sets. We tune our hyperparameters on the validation sets.

2.6.3 Evaluation Metric

We use the mean average precision (mAP) among the average precision (AP) on each of the seven surgical tools, which is the same as the challenge evaluation metric. To ensure a fair comparison with all the methods during and after the challenge, we exactly follow every detail of data usage and evaluation protocol used in M2CAI challenge.

2.6.4 Experimental Results

2.6.4.1 M2cai-tool dataset.

In this experiment, we choose the winner's and the 3rd place's methods from the challenge, as well as three approaches after the challenge as comparison methods. Among the challenge methods, EndoNet [4] first proposed using CNN as a baseline model. The winner of the challenge [19] introduced an ensemble model of VGGNet and Inception Net. However, the highest mAP is a little above 60%. For the methods after the challenge, both Jin *et al.* [50] and Choi *et al.* [48] added location information of the tools by adding surgical tools bounding box to the dataset. These two approaches improved the mAP by 10%. AGNet [49] proposed to use an attention model to increase the detection performance. AGNet trained two cascaded deep convolutional neural networks: the first one as a global model to locate the area which has higher responses by the attention based classification network, and then the second one as a local model to classify the cropped areas with higher attention. Before our method, AGNet has the best mAP among all the approaches. We compare all these methods with our results of STGCN results. We include three variations of the proposed STGCN as side ablation experiments. STGCN (DenseNet) is the model we train and test on the labeled images without using any temporal information. STGCN (3D DenseNet + LSTM) contains the inflated 3D DenseNet as the backbone, and add an LSTM layer after it to extract the temporal information from continuous frames in the video. The difference between STGCN (3D DenseNet + GCNs) and STGCN (3D DenseNet + LSTM) is that STGCN (3D DenseNet + GCNs) uses GCNs to exploit the temporal information.

As shown in Table 2.9, the STGCN (DenseNet) model has achieved better performance than all existing methods. Compared to AGNet, STGCN (DenseNet) has

Methods	Mean AP
STGCN (3D DenseNet + GCNs)	**90.24**
STGCN (3D DenseNet + LSTM)	89.03
STGCN (DenseNet)	88.27
AGNet [49]	86.8
Choi et al. [48]	72.3
Jin et al. [50]	71.8
Sheng et al. [19]	63.8
Twinanda et al. [4]	52.5

Table 2.4. The results on M2cai-tool dataset.

not used any attention strategy to boost the performance to have around 2% better mAP than AGNet. By adding temporal information, the STGCN (3D DenseNet + LSTM) and the proposed STGCN (3D DenseNet) both improves our image classification model STGCN (DenseNet). With GCNs, it could have 1% better mAP than LSTM. Our results demonstrate that temporal information is effectively helpful for surgical tool presence detection, and GCNs is better than LSTM in this problem.

2.6.4.2 Cholec80 dataset.

We compare the proposed STGCN result with the two baseline methods ToolNet and EndoNet on this dataset in [4]. We also try the four different temporal pooling methods: l_2 pooling (STGCN(l_2)), average pooling (STGCN(avg)), max pooling (STGCN(max)), and max-min pooling (STGCN) on this dataset. Results are shown in Table 2.10. On this larger dataset, the proposed STGCN has better performance than the baseline methods ToolNet and EndoNet modeling the problem as a multi-label image classification problem. By utilizing the temporal information, the proposed STGCN has improved the performance around 10% in mAP.

	ToolNet [4]	EndoNet [4]	STGCN (l_2)	STGCN (avg)	STGCN (max)	STGCN
mAP	80.9	81.0	90.05	90.11	90.08	**90.13**

Table 2.5. The results on Cholec80 dataset.

Among all the results with different temporal pooling strategies, max-min pooling has better performance. However, the improvement is so small that it could be caused by randomness during model training. The slight difference among the four pooling methods offers support to our analysis that the graph convolutional layer has utilized the temporal information so how to aggregate the information along the temporal dimension is not sensitive, which could be convenient for model designing.

By comparing the results of the proposed STGCN with the existing methods on both M2cai-tool and Cholec80 datasets, it demonstrates that there is always significant improvement by utilizing the extra temporal information by modeling the surgical tool presence detection as a video classification problem. Besides, with the power of GCNs, STGCN has better accuracy even compared with existing leading methods using multiple CNNs [49] or labeling additional localization ground truth [50].

2.7 Methodology Three: A Semi-Supervised Framework for Video Classification

2.7.1 Model Overview

As shown in Figure 2.5, the proposed semi-supervised learning framework consists of two encoders: the spatial encoder and the temporal encoder. The spatial encoder extracts the spatial features of the video segment, including one central labeled frame and its neighbor unlabeled frames. Then the temporal encoder extracts the temporal information from the continuous frames and generates the final spatial-temporal representation for surgical tool detection.

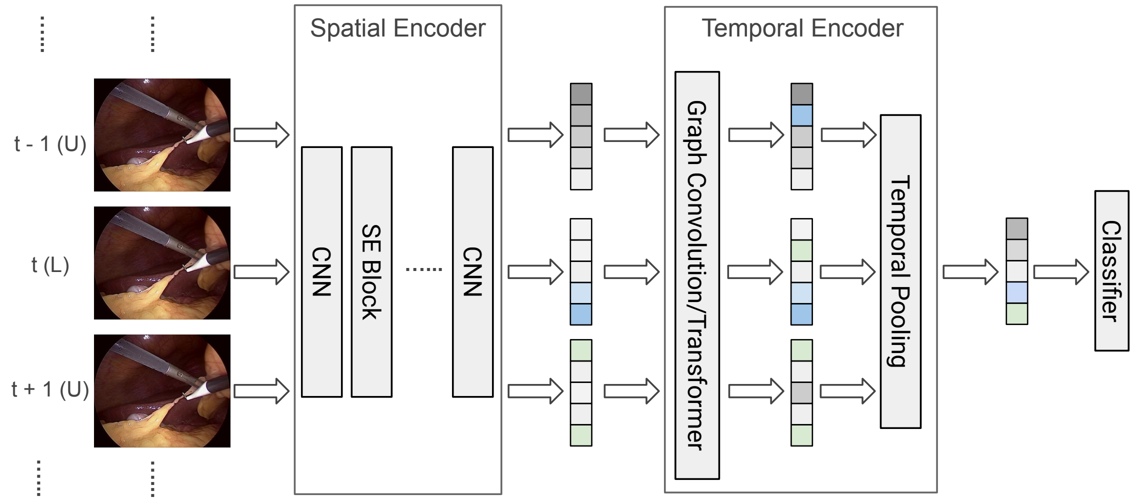


Figure 2.5. The overview of the proposed semi-supervised framework for surgical tool presence detection. L is short for labeled and and U for unlabeled..

2.7.2 Spatial Encoder

The spatial encoder aims to extract high-level features of each image. Though any type of deep neural networks could be used as the spatial encoder, CNNs are naturally good fits for images classification problems and have the best performance so far on different image classification benchmarks [83, 84, 85, 23, 86]. We explore performances of different state-of-the-art models including ResNet [83], DenseNet [23], MobileNet [86], SENet [85] and PNASNet [84] on surgical tool detection problem. ResNet proposes to represent the data with residual learning via shortcut connections. The residual learning could help to alleviate the gradient descent problem in very deep neural networks and lead to powerful data representations. DenseNet connects each layer to every other layer in a feed-forward fashion to encourage feature reuse and strengthen feature propagation. SENet proposes a channel-attention module to choose which channels to focus on adaptively. PNASNet is a very complex and deep model by structure searching with machine learning techniques. It has the best performance on several large-scale benchmarks. To explore if a smaller model (fewer parameters and fewer computations) is a good fit for the surgical tool detection problem, we include MobileNet, a small and compact models having good performance on mobile devices, in our experiments. We finally use SENet as the backbone of the spatial encoder. Detailed experiments and explanations are listed in Section 4.2.

2.7.3 Temporal Encoder

After the spatial encoder, the temporal encoder takes the spatial features of all the frames in the input video segment. It first adaptively extracts the temporal information and encodes the spatial features with temporal information, then generates the final spatial-temporal representation for the whole input video segment. For the temporal encoder, we propose two models: one with graph convolution networks, which leads to the proposed model ST-GCN and the other with the temporal attention module and the model ST-TAN. The Graph Convolution and temporal pooling are the same as the descriptions in the method two.

2.7.3.1 Temporal Attention

Self-attention proves to be successful in natural language modeling [46, 26]. BERT [46] is a general model that is pre-trained in an unsupervised fashion and has good performance on various language modeling tasks. The basic module of BERT model is the Transformer layer. The Transformer layer has three components: a pre-attention shared fully-connected layer, a self-attention module, and a post-attention shared fully-connected layer. The attention mechanism [26] in the Transformer encoder is the scaled dot-product attention. It maps the input data into three parts, a query matrix, a key matrix, and a value matrix. The query matrix works together with the key matrix to serve as the input of the Softmax. Then the Softmax function creates the attention weights, which will be later applied to the value matrix to generate the output features with the attention on the whole sequence. The self-attention layer is formulated as:

$$Z = Softmax\left(\frac{\left(XW^{Q}\right)\left(XW^{K}\right)^{T}}{\sqrt{d}}\right)XW^{V}, \tag{2.7}$$

where $X \in R^{N\times d}$ is the input spatial feature matrix, W^{Q}, W^{K}, and $W^{V} \in R^{d\times d}$ corresponds to the query, key, and value weight matrix. $\sqrt{d}$ is a scaling factor, and Z is the output of the attention layer.

2.8 Experiments

2.8.1 Implementation Details

2.8.1.1 The Spatial Encoder

To explore different structures as the spatial encoder in our semi-supervised framework, we have trained different CNNs, including ResNet, DenseNet, MobileNet, SENet, and PNASNet. Specifically, we use ResNet-101, DenseNet-121, MobileNetV2, SENet-154, and PNAS-5-Large in model training. Different models required different input image shape. For PNASNet-5-Large, we preprocess the image to the size of 331×331. For other models, we use the image with a size of 224×224. All the models are pretrained on ImageNet krizhevsky2012imagenet dataset and trained (finetuned) on M2cai-tool and Cholec80 training datasets with Adam optimizer with initial learning rate 0.0001 for 50 epochs. The learning rate is decayed to its 0.95 if the training loss does not decrease after three continuous training epochs.

2.8.1.2 The Temporal Encoder

The temporal encoder takes the spatial features from the spatial encoder as input. We extract the features from the second-to-last fully-connected layer of the spatial encoder as spatial encoder features. The spatial features size of SENet is 2048. In the temporal encoder of the proposed ST-GCN, a single graph convolutional layer is introduced to encode the temporal information. In ST-TAN, we apply one Transformer layer to encode the temporal information. The Transformer layer first maps the spatial features from the dimension of 2048 into the dimension of 1024. Then the temporal information is learned inside the temporal attention module. We use a 4-head self-attention module here. After the temporal information is encoded, one temporal pooling layer is added to pool the features along the temporal dimension. Following that is a fully-connected layer mapping 1024 dimensions feature to the number of surgical tools for classification. In GCNs, both batch normalization and dropout are added after the graph convolutional layer. We train the ST-GCN and ST-TAN with Adam optimizer with an initial learning rate of 0.0001 for 100 epochs. The dropout rate is set as 0.75 in our training. The same learning rate decay strategy is used as the one in training spatial encoders. For max-min pooling, we fix the hyperparameter α to 0.75.

2.8.2 Data Description

We use the same data as our method two. In the following experiments, we have all our ablation studies on the M2CAI-tool dataset.

2.8.3 Ablation Studies

Our ablation studies all use M2cai-tool datasets. The ablation studies include the importance of image alignment, model performances between different spatial encoder structures, performances of different graph similarities, performances of different temporal pooling methods, and length of the unlabeled frames to use.

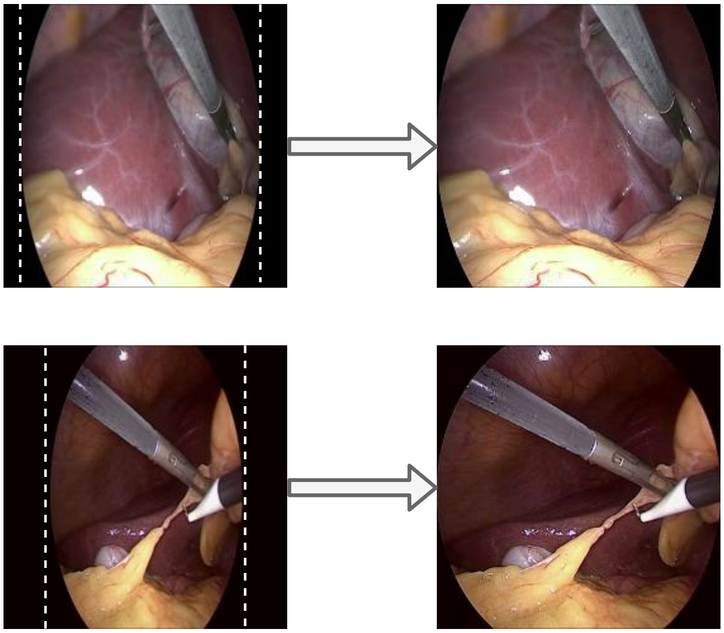


Figure 2.6. Left: the original frames in M2cai-tool dataset with different ratios of black boarder. Right: the frames after image alignment..

2.8.3.1 Image Alignment

The surgical videos captured by the laparoscope always contains black borders with zero gradients. It happens in both M2cai-tool and Cholec80 datasets, as shown in Figure 2.6. If the original datasets are used for model training, the performance of the model would not as good as the datasets after image alignment. It is because the black board contains no information but could make the statistical distribution of different video images vary a lot, making the model training more difficult. To verify the effect of the image alignment, we train two MobileNetV2 models on the original M2cai-tool dataset and the dataset after image alignment. All hyperparameters and training settings of the two models are the same except for the datasets. The validation mAP on the original dataset is 80.25, and on the aligned dataset is 83.49. Thus, we apply image alignment on both datasets for the following experiments.

2.8.3.2 The Spatial Encoder

To compare the performances of different state-of-the-art image classification models in surgical video detection, we train the ResNet-101, DenseNet-121, MobileNetV2, SENet-154, and PNASNET-5-Large. As shown in Table 2.6, these spatial models vary among different sizes of parameters and different computation costs. PNASNet-5-Large has got the best performance 89.80 mAP, so we use PNASNet as the spatial encoder for the M2cai-tool dataset. However, we use SENet-154 as the spatial encoder for the Cholec80 dataset. It is because the input image size for PNASNet is 331, which is very memory consuming. Considering the Cholec80 is much larger than M2cai80 (65 more videos), it is reasonable to use a smaller model. MobileNet has the lowest mAP while it is much lesser time and space complexity than other models. It might be a good fit when the computational resource is limited. ResNet and DenseNet have similar performances.

2.8.3.3 Graph Similarities

In ST-GCN, we need to build the similarity graph from the spatial features of the video segment. There are different similarity metrics that we could use to build the similarity graph. We include the comparison of building the similarity graph with cosine similarity, l_1 similarity, l_2 similarity, and Chebyshev similarity. Each model is

Model Name	Params(M)	FLOPS(G)	Image	Spatial Size	mAP
ResNet-101	44.55	7.87	224	2048	88.23
DenseNet-121	7.98	2.90	224	1024	88.27
MobileNetV2	3.51	0.33	224	1280	83.49
SENet-154	115.09	41.72	224	2048	**89.56**
PNASNet-5-Large	86.06	25.20	331	4320	**89.80**

Table 2.6. The comparison of different models as our spatial encoder.

Similarity Type	Cosine	l_1	l_2	Chebyshev
mAP	**90.86**	90.83	90.82	90.83

Table 2.7. The results of using different similarity metrics to build similarity graph.

trained on video segments of 15 frames for 100 epochs. The results are shown in Table 2.7. With cosine similarity, the highest mAP 90.86 is achieved. There are only slight differences between different ways to build the similarity graph, and the differences might be caused by the randomness in model initialization or training process. Since the similarity metric is not sensitive to the final performance, we could use any metric as the component in the proposed ST-GAN. In the following experiments, we use cosine similarity to compute the graph.

2.8.3.4 Temporal Pooling Methods

After the graph convolutional layer or Transformer layer, we get the temporal information encoded features for every frame in the input video. To get the final prediction, we need to combine these features into a single representation for the input video. We apply temporal pooling here. In this experiment, we analyze how different pooling methods contribute to the final detection performance. We use the spatial features from SENet and keep the length of input videos as 15. All other training settings are the same for l_2, max, average, and min-max pooling. As shown in

Temporal Pooling Type	l_2	Max	Average	Min-Max
mAP	90.85	90.83	90.84	**90.88**

Table 2.8. The ablation study of using different temporal pooling methods on M2cai-tool dataset.

Table 2.8, there is not much difference between different pooling method. It is similar to the ablation study for similarity metrics. It might because the temporal feature could be easily captured and encoded in the graph convolutional layer or Transformer layer. The pre-processing (building similarity graph) and post-processing (temporal pooling) are not sensitive. For the rest of the experiments, we use min-max pooling as our standard temporal pooling method in both ST-GCN and ST-TAN.

2.8.3.5 Length of the frames.

Since we use the labeled frame and its neighbor unlabeled frames as the input video segment, we would like to see if there is a relationship between the model performance and how many unlabeled frames we use. We start from SENet spatial features and train ST-VTN models with different video lengths (one labeled frame and others unlabeled). For each model, we use the same training settings except for the video length and train the model for 100 epochs. The model performances with different video lengths from 1 to 41 are shown in Figure 2.7. When the video length is 1, it means the model training uses the labeled images only. The performance is very similar to when we only use the spatial encoder for training, as shown in Table 2.6. When the video length is larger than 1, it includes unlabeled frames as a part of training data. As shown in Figure 2.7, when we use more unlabeled frames, the overall mAP keeps increasing. It is beneficial for introducing unlabeled frames to add temporal information. However, it is not reasonable to use a very long video length because the computation costs could be incredibly high. We use the length of 41 on the M2cai-tool dataset. For the Cholec80 dataset, we use the video length of 19 because the dataset is much larger than M2cai-tool.

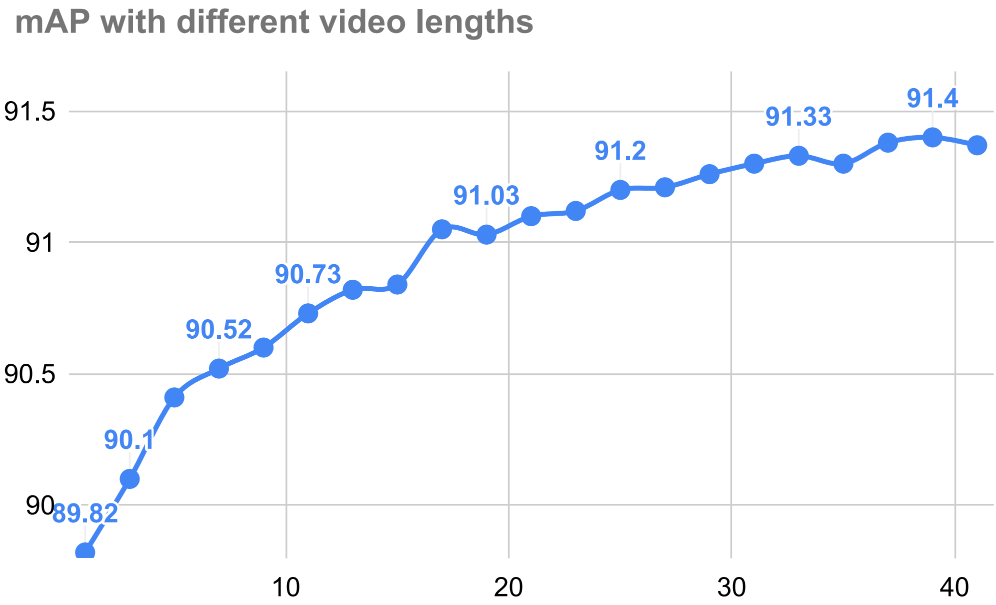


Figure 2.7. The performances on M2cai-tool dataset with different video lengths..

2.8.4 Experimental Results

2.8.4.1 M2cai-tool dataset

To give a fair comparison with state-of-the-art methods for surgical tool presence detection on the M2cai-tool dataset, we include the comparison with methods from the M2CAI tool presence detection challenge twinanda2016endonet,wang2017isbi and methods after the challenge choi2017surgical,hu2017agnet,jintool,sahu2017addressing. As shown in Table 2.9, we also include more information about the models that if the model needs extra labels, uses temporal information, and is a semi-supervised model. For the models in the challenge, we choose the winner's wang2017isbi and the 3rd place's twinanda2016endonet methods. EndoNet twinanda2016endonet first proposed using CNN as a baseline model for the surgical tool detection problem. The winner of the challenge wang2017isbi introduced an ensemble model of VGGNet and InceptionNet, along with a few data engineering like data augmentation. However, the highest mAP was no larger than 0.65. After the challenge, ZIBNet focused on alleviating the data imbalance problem and slightly improved the performance to 0.65. Since solving the surgical tool localization problem would benefit surgical tool presence detection, both [48] and [50] introduced extra surgical tool localization information (bounding boxes labels) to the original dataset, then solve the problem with general object detection models. The model from [48] improves the performance by 7.3%. The model from [50] further improved the mAP by 9.5% with a more powerful object detection model. AGNet hu2017agnet does not use extra localization information. AGNet trained two cascaded deep convolutional neural networks: the first one as a global model to locate the area which had higher responses by the attention-based classification network, and then the second one as a local model to classify the cropped areas with higher attention responses. AGNet has the best mAP among all existing approaches. In Table 2.9, we include MTRCNet-CL jin2020multi since it does has competing performance compared to state-of-the-art methods. It utilized the temporal information and the correlation between surgical tool detection task and surgical phase recognition task. However, the M2cai-tool dataset does not contain extra surgical phase labels, and we are not able to train MTRCNet-CL on the M2cai-tool dataset. We compare all these methods with the proposed semi-supervised learning framework. We include the three proposed models as side ablation experiments. ST-SPN is the model with PNASNet as the spatial encoder. It is a supervised model since it contains not a temporal encoder. We train and test on the labeled im-

Methods	Extra Label	Temporal	Semi-supervised	mAP
Twinanda et al. [4]				52.5
Wang et al. [19]				63.8
ZIBNet [63]				65.0
Choi et al. [48]	$\surd$			72.3
Jin et al. [50]	$\surd$			81.8
AGNet [49]				86.8
MTRCNet-CL [10]	$\surd$	$\surd$		N/A
ST-SPN				89.8
ST-GCN		$\surd$	$\surd$	**91.33**
ST-TAN		$\surd$	$\surd$	**91.38**

Table 2.9. The results on M2cai-tool dataset.

ages without using any temporal information. ST-GCN contains SENet as the spatial encoder and graph convolutional networks as the temporal encoder. ST-TAN shares the same spatial encoder with ST-GCN, while the temporal encoder is a Transformer layer focusing on how to use the temporal information with the self-attention module adaptively.

As shown in Table 2.9, ST-SPN has achieved better performance than all existing methods, and it has improved 3% mAP compared to the best performance achieved by AGNet. It even largely outperforms the models with extra tool localization labels choi2017surgical,jintool. The improvement comes from two parts. First, the backbone of the ST-SPN is PNASNet, a powerful model structure found by model searching with machine learning. Second, we include the image alignment in the data pre-processing stage and data augmentations in the training stage. After adding the temporal encoder, the performances of both ST-GCN and ST-TAN are further improved. The ablation study with different video lengths has shown that including more temporal information would benefit in solving the surgical tool detection problem. The proposed ST-GCN and ST-TAN not only use temporal information but also utilize the temporal information from unlabeled part of surgical videos. The two models have gained about 2% mAP compared with ST-SPN. Since the performance of ST-GCN is close to that of ST-TAN, we can conclude that both ST-GCN and ST-TAN are a good fit to extract useful information for surgical tool detection from labeled and unlabeled data.

2.8.4.2 Cholec80 dataset

For the Cholec80 dataset, we compare the proposed method with several well-known and state-of-the-art methods. We include a deformable part model (DPM) reported in [10]. The DPM model consists of three components to model each tool and use HOG features to represent the surgical images. ToolNet and EndoNet twinanda2016endonet have been proposed along with the Cholec80 dataset as the benchmarks. ToolNet is the first CNNs model solving the surgical tool detection problem, and EndoNet adds the surgical phase label as one additional feature to help the detection. We also compare our methods to MTRCNet-CL jin2020multi, which has the best performance on Cholec80 so far. MTRCNet-CL models the detection problem along with the surgical phase recognition problem as a multi-task learning problem. We list the result of the proposed ST-TAN here to compare with these existing methods. For our ST-TAN, we use the video length of 19 in training and evaluation.

As shown in Table 2.10, DPM could not perform well on all the surgical tools like scissors and irrigator. The data imbalance and the low-level HOG features are the main reason for the failure of DPM. As the features are learned with CNNs in ToolNet and EndoNet, the performance is improved by more than 20%. The mAPs of ToolNet and EndoNet are very similar. Though EndoNet aims to improve the surgical tool detection performance with the help of surgical phase information,

	DPM	ToolNet	EndoNet	MTRCNet-CL	ST-TAN
Grasper	82.3	84.7	84.8	84.7	**88.9**
Bipolar	60.6	85.9	86.9	90.1	**90.6**
Hook	93.4	95.5	95.6	95.6	**95.8**
Scissors	23.4	60.9	58.6	86.7	**89.9**
Clipper	68.4	79.8	80.1	89.8	**90.7**
Irrigator	40.5	73.0	74.4	88.2	**88.7**
Specimen bag	40.0	86.3	86.8	88.9	**90.4**
Mean	58.4	80.9	81.0	89.1	**90.7**

Table 2.10. The results on Cholec80 dataset.

EndoNet is only 0.1% better than ToolNet. It is reasonable since EndoNet only takes the phase label as one additional feature to the features in ToolNet. After modeling the tool detection problem and phase recognition problem together and solve them as a multi-task problem, MTRCNet-CL improves the performance by 8% and has a better performance on each tool. Our proposed ST-TAN performance 1.6% better on mAP than MTRCNet-CL even we do not use extra phase recognition labels. Our ST-TAN also has better single tool detection performance than MTRCNet-CL.

2.9 Conclusion

Surgical tool presence detection is an essential problem for automatic surgical video analysis. To use the temporal information from the video data, we propose a novel model named STGCN which applies graph convolutional learning on continuous video frames to better use the temporal information. STGCN can directly take a video (a sequence of image frames) as input, extract both spatial and temporal features of the input and get excellent surgical tool detection precision. To the best of our knowledge, this is the first model which can take video sequences as inputs for surgical tool presence detection. On both of the two datasets to evaluate our model, STGCN has the best mean average precision. Comparing with the models that only use spatial features, we demonstrate that with GCNs, the temporal information is effective to improve surgical tool presence detection performance.

CHAPTER 3

Molecule Property Prediction with Large Scale Unsupervised Pre-training

3.1 Introduction

The capability of accurate prediction of molecular properties is an essential key in the chemical and pharmaceutical industries. It benefits various academic areas and industrial applications such as improvement to rational chemical design, reducing R&D cost, decreasing the failure rate in potential drug screening trials, as well as speeding the process of new drug discovery [20]. The key problem of introducing Deep Learning into this area lies on embedding graph-like molecules onto a continuous vector space. Then the representations, as named molecular fingerprints, could be used for various applications such as molecular properties classification, regression, or generating new molecules. Instead of computing a basic property, traditional molecular fingerprints provide a description of a specific part of the molecular structure rogers2010extended. However, traditional molecular fingerprints require intensive manual feature engineering and strong domain knowledge. Besides, this kind of fingerprints is highly task-dependent, not general enough for other property prediction tasks [22].

The current success of deep learning in various areas and applications, e.g., image classification [23, 25], video understanding [26, 27, 5], medical imaging [28, 29, 12], and bioinformatics [30, 32], demonstrates that deep learning is a powerful tool in learning feature from data and good at task-related prediction. An increasing number of publications have introduced deep learning into molecular fingerprint learning [30, 32, 39, 40]. The models being introduced rely on two main deep learning structures: Recurrent Neural Networks (RNNs) [45] and Graph Convolutional Networks (GCNs) [44, 65]. For RNNs-based methods, molecules are represented as strings by Simplified Molecular-Input Line-Entry system (SMILES). In this way, the current successful models in natural language modeling could be utilized to extract high-quality features from SMILES and make task-related predictions. GCNs-based methods consider the atoms in molecules as graph nodes and the chemical bonds as graph edges. These methods use graph convolutions to extract the feature then classify/regress the molecular properties. In general, it is not trivial to support RNNs-based methods for parallel training on multiple GPUs and multiple devices, and it needs different training tricks like gradient clipping and early stopping to assure the model convergence; GCNs-based methods usually have high computation complexity. It limits exploring more complicated methods for molecular properties prediction. Meanwhile, CNNs-based models [87, 26] for language translation and modeling have been developed and widely used. These methods could easily support parallel training. With the help of attention mechanism, the results even outperform a lot of RNN models.

The success of current deep learning methods highly relies on a large-scale labeled training samples. For many areas, the labeled sample number of image classification could easily reach several million or more. However, it is not the same situation with molecular property prediction. The cost of obtaining such scale of molecular properties with screening experiments is exceptionally high. It is similar to the case in natural language modeling that they have almost unlimited unlabeled data while a tiny portion has labels. The state-of-the-art framework to utilize the unlabeled data is the pre-training and fine-tuning framework [41]. It pre-trains the model in an unsupervised fashion then fine-tune the model on labeled data. Seq3seq Fingerprint model [32] first starts using this framework to involve large-scale unlabeled data in model training to improve the prediction performance. However, Seq3seq model is not very efficient since it uses an encoder-decoder structure, and the decoder is used as a scaffold and does not contribute to the final prediction.

The motivations of this paper are two-folded. First, we would like to build a powerful semi-supervised model utilizing the essential information in unlimited unlabeled data to improve the prediction performance with limited labeled data. Second, we would like our model to be efficient in training stage in two ways: 1) our model should naturally support parallel training to reduce pre-training time; 2) the model used for pre-training will all take part in the fine-tuning stage with no scaffolding part like the decoder of Seq3seq fingerprint [32]. Thus, in this paper, we propose a pre-training and fine-tuning two-stage framework named SMILES-BERT motivated by the recent natural language modeling work BERT [46]. The neural network structure is a fully convolutional net stacked of Transformer layers. In the pre-training task, SMILES-BERT is trained with unsupervised learning mechanism Masked SMILES Recovery on large scale unlabeled data. In the Masked SMILES Recovery task, the input SMILES will be randomly masked or corrupted, and the model is being trained to recover the original SMILES according to the information lying in the unmasked part of the input. After that, the model needs a slight fine-tuning with the labeled dataset to have good prediction performance. The proposed SMILES-BERT contains several benefits than the existing methods: 1) different from Seq2seq or Seq3seq model, SMILES-BERT does not require an encoder-decoder structure which is more efficient and the model could be more complicated given the same GPU memory; 2) SMILES-BERT is more natural to parallel training because of the fully convolutional structure; 3) The random masking method will having SMILES-BERT more general and able to avoid overfitting; 4) The attention mechanism is used in the Transformer layer which could potentially improve the prediction performance.

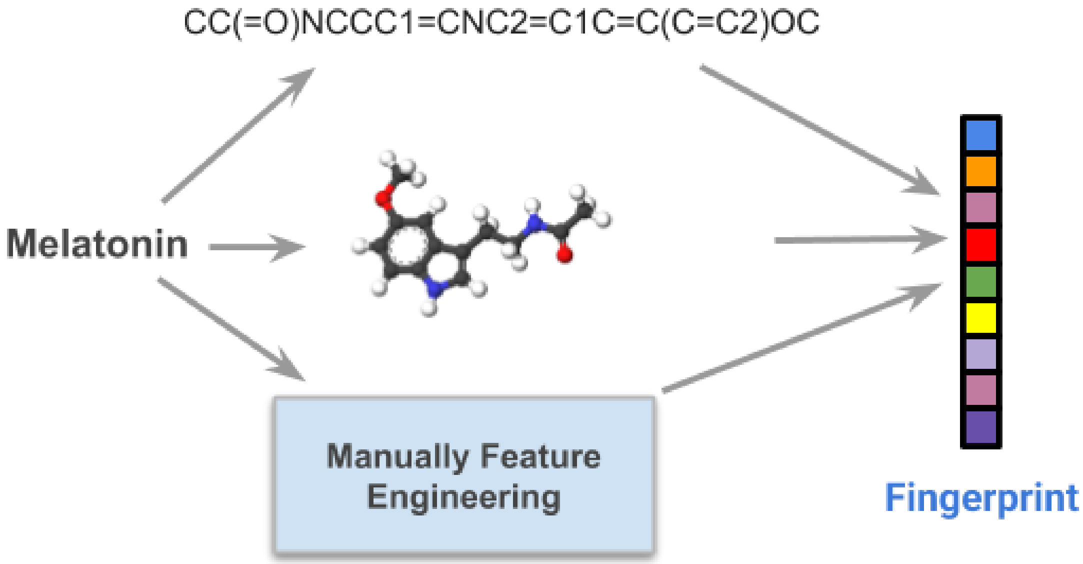


Figure 3.1. Mapping molecule to feature vector (Fingerprint) with different methods..

Our contributions of this paper could be summarize as:

- We propose a two-stage (pre-training and fine-tuning) model SMILES-BERT [33] to utilize both unlabeled data and labeled data to have better molecular properties prediction performance.
- SMILES-BERT has better performance, outperforming a series of state-of-the art methods on three datasets.

3.2 Related Work

Almost all the molecular property prediction methods or fingerprints could be concluded in Figure 3.1. The most important task is to embed the molecule into a continuous feature space for further task. Since molecules have different representation, these methods could be divided into three categories based on the input representation format being used: the manually feature engineering methods, the graph-based methods, and the sequence-based methods.

3.2.1 SMILES and canonical SMILES

To represent molecules with atoms and chemical bonds inside, the Simplified Molecular-Input Line-Entry system (SMILES) [88] is proposed to represent molecules in a simple way. SMILES is a line notation which 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. One example of SMILES representation is shown in Figure 3.1: melatonin with structure $C_{13}H_{16}N_2O_2$, where corresponding SMILE representation is included as well as the 3D molecule structure. Simply speaking, the letters, e.g., C, N, generally represent the atoms, while some symbols like $-, =, \#$ represent the chemical bonds. SMILE system is not perfect given that the vanilla SMILE system is not a bijective mapping between SMILE sequence and a molecule. For example, a molecule could have multiple corresponding SMILE representations, e.g., CCO, OCC and $C(O)C$. To address this issue and provide a one-to-one mapping between SMILES and molecules, multiple canonicalization algorithms are invented to ensure the representation uniqueness of each molecular structure [89]. In this paper, all the SMILES are canonical.

3.2.2 Manually Designed Fingerprint

Traditionally, there is a class of molecular representation systems called molecular fingerprints. 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 have invented new fingerprint systems which can benefit future predictive tasks.

Many hash-based methods has been proposed to generate unique molecular feature representation [90, 22, 91]. 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 previous layer. One of the most famous ones is Extended-Connectivity FingerPrint (ECFP) [21]. However, due to the non-invertible nature of the hash function, the hash-bashed fingerprint methods usually do not encode enough task-related information and hence result in not good enough performance in properties prediction.

Another stream of traditional fingerprint methods are based on the biological experiments and the expertise knowledge and experience, e.g., [92, 93]. Biologists have figured out several important task-related sub-structures (fragments), e.g., $CC(OH)CC$ for solubility prediction, and count those sub-structures as local features to produce molecular fingerprints. This kind of fingerprint methods usually work well for specific tasks, but could not generalize well for other tasks.

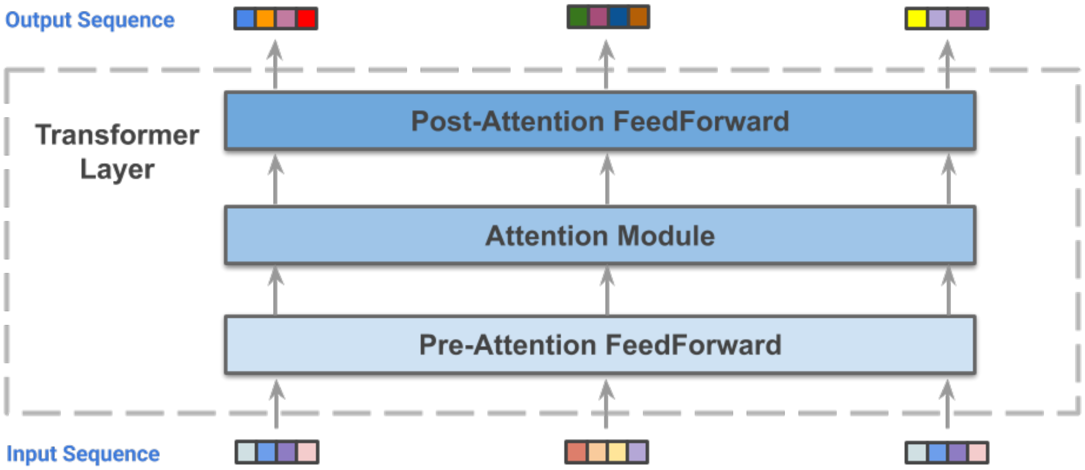


Figure 3.2. The structure of Transformer Layer..

3.2.3 Deep Fingerprints

The growth of deep learning has provided excellent flexibility and performance to learn molecular fingerprints from data samples, without explicit guides from experts [87, 94, 95, 96, 97, 30].

3.2.3.1 **Graph-based fingerprint**

Among all the graph-based molecular fingerprint, the state-of-the-art work is the neural fingerprint [39]. The neural fingerprint mimics the whole process of generating circular fingerprint but the hash function is replaced by a non-linear activated densely connected layer. The model of neural fingerprint is a 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.

3.2.3.2 **RNNs-based fingerprints**

Recently, a few unsupervised fingerprint methods, e.g., seq2seq fingerprint [30], are proposed to alleviate the issue of 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 any kind of classifiers. 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 with prediction tasks, meaning that the representation only adjusts to the recovery task of the original raw representation. It might not provide optimal inference performance for general prediction task. Seq3seq [32] is the first semi-supervised learning model for molecular property prediction. It has an Encoder-Decoder structure which could learn the fingerprints based on self-representation. Thus, it could utilize unlimited unlabeled data. However, the Encoder-Decoder framework limits its capability for property prediction. It is because the decoder of Seq3seq functions as a scaffold in pre-training stage and is barely useful in fine-tuning, but it has to consume the GPU memory in the pre-training stage. In this way, Seq3seq fingerprint is not computationally effective.

3.2.4 Transformer and BERT on Natural Language Modeling

Recently, there are several CNNs-based language models having excellent performance on various language modeling tasks [98, 99, 26, 100, 101, 46]. These methods use fully convolutional network structures instead of any RNNs blocks. With the help of self-attention mechanism [26], CNNs-based models could even outperform RNNs-based models. Among these methods, Transformer [26] is one of the most significant model building block. Furthermore, BERT [46] proposes to pre-train the Transformer encoders with two tasks: masked language learning, and continuous sentence classification. Both Transformer and BERT belongs to pre-train and fine-tuning framework, which could use the power of unlabeled data to initialize the parameters in the models, then promise good performance in following general language modeling tasks. This paper is inspired by Transformer and BERT, we keep the model used in BERT as our backbone with a few adaptations.

3.2.5 Seq3seq fingerprint

The Seq3seq fingerprint [32] is our previous work. Different from traditional models [39, 30], 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 **??**. The semi-supervised training is done by two tasks: the self-recovery task and the inference task. The whole pipeline is illustrated in Figure 3.3.

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 3.3, this task contains a perceiver network and an interpreter network.

This structure is motivated by the seq2seq model [30, 45]. The original seq2seq model is used in machine translation [45]. 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 [30] 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 [39, 30] 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 [30]. 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 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.

3.2.6 End-to-end Semi-supervised Learning

As shown in Figure 3.3, the semi-supervised loss $\mathcal{L}_{semi}$ combines the unsupervised loss $\mathcal{L}_{unsup}$ and the supervised loss $\mathcal{L}_{sup}$ together as

$$\mathcal{L}_{semi} = \mathcal{L}_{unsup} + \lambda \mathcal{L}_{sup}. \tag{3.1}$$

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 [30]. 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 [30].

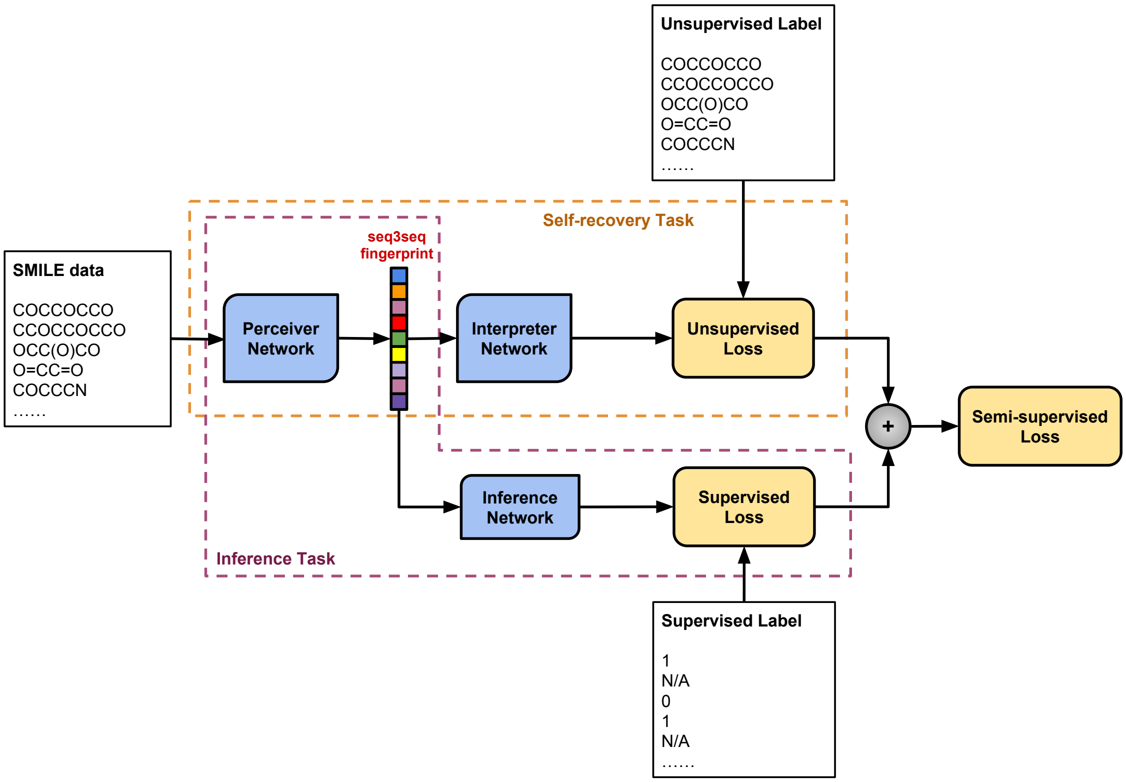


Figure 3.3. This figure shows the proposed seq3seq fingerprint model. The proposed model is trained through two tasks: a self-recovery task and an inference task. The self-recovery task contains a perceiver network and an interpreter network; the inference task shares the perceiver with self-recover task and has an inference network. The semi-supervised loss is the sum of supervised loss and unsupervised loss..

3.3 Methodology: SMILES-BERT

In this section, the proposed SMILES-BERT is introduced step by step. First, we give the details of our backbone and its building block, i.e. Transformer Encoder. Then the Masked SMILEs Recovery task used for pre-training our backbone on large scale unlabeled data will be introduced. Following that is the fine-tuning process for molecular property prediction. The proposed model handles the molecules as sequences. Thus the inputs of SMILES-BERT are the tokenized molecules SMILEs representations as shown in Figure3.4.

3.3.1 Model Backbone and Transformer Layer

As shown in Figure 3.2, a transformer layer contains three components: a pre-attention feed forward neural network, a self-attention layer, and a post-attention feed forward neural network. The pre-attention feed forward is a fully-connected layer shared by all the input tokens. It maps the output feature from former Transformer layer or the embedded feature from the input into another nonlinear space. The post-attention works precisely in the same way, while the input is the output features after self-attention module.

RNNs-based methods utilize the sequential information naturally since the output from the former time step will be part of the input of the current time step. However, in Transformer Encoder, only using feed forward network could not bring temporal information from the sequence. The self-attention layer plays a crucial role to introduce the temporal relation into consideration for feature learning. For every time step, it could decide how to use information from other sequences by which is more related to itself.

The attention mechanism [26] used in Transformer Encoder is named scaled dot-product attention. It maps the input data into three parts, a query matrix, a key matrix, and a value matrix. The query matrix works together with the key matrix to serve as the input of the softmax. Then softmax creates the attention weights, which will be applied to the value matrix to generate the output features with the attention on the whole sequence. The scaled doc-product attention is formulated as:

$$Z = Softmax\left(\frac{\left(XW^{Q}\right)\left(XW^{K}\right)^{T}}{\sqrt{d_k}}\right)XW^{V},$$

Where $X \in R^{N \times M}$ is the input feature matrix, W^{Q}, W^{K}, and $W^{V} \in R^{M \times d_k}$ corresponds to query weight matrix, key weight matrix, and value weight matrix. $\sqrt{d_k}$ is a scaling factor and Z is the output of the attention layer. It is the single head self-attention used in BERT. However, in the backbone, a more powerful version of the self-attention layer is used, the multi-head self-attention. Thus, different heads could pay attention to various aspects, making attention to the best power.

All the three components, the feed forward neural networks, and the self-attention layer are following by a normalization layer to increase the generalization ability of the model. Besides, each of the components has a residual input to better utilize the original information.

The whole structure of the proposed model is shown in Figure 3.4. SMILEs BERT contains a stack of Transformer Encoders with the self-attention mechanism.

3.3.2 Pre-training as Masked SMILEs Recovery

The pre-training stage is shown in Figure 3.4. BERT uses a combination of two tasks to per-train the model, masked language learning and the consecutive sentences classification. Masked language learning is that given a partially masked sentence, using other visible works to predict the masked parts. It is label-free so it could utilize all the unlabeled sentence in natural languages. The consecutive sentences

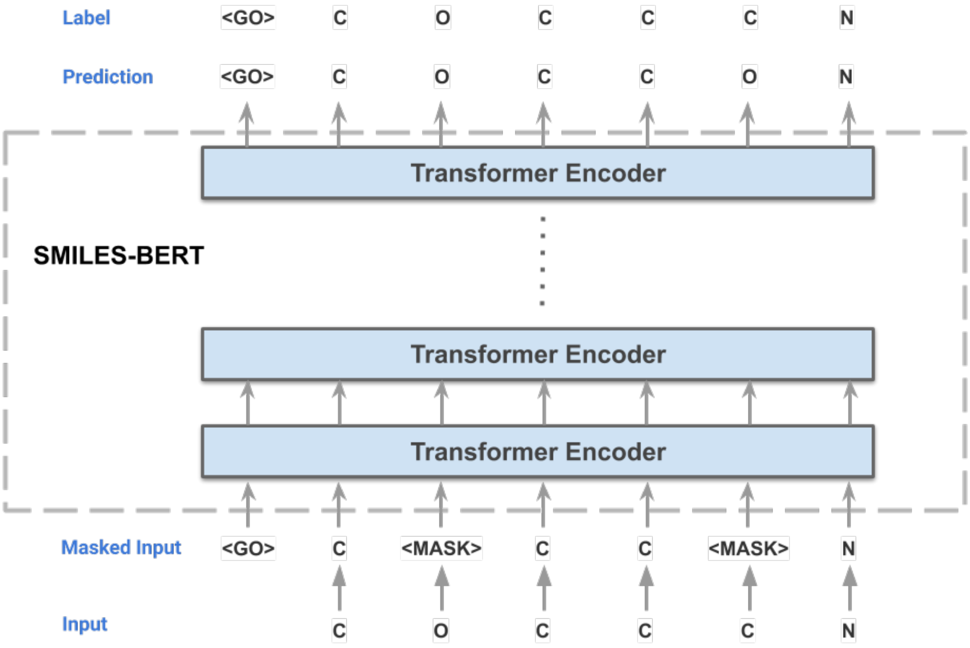


Figure 3.4. SMILES-BERT: pre-training stage..
smiles

classification is to classify if two sentences are consecutive, which is also label-free. However, different from natural language modeling, SMILEs do not have a consecutive relationship. The masked language learning is still promising in pre-train the model with unlabeled SMILEs and we name the task Masked SMILEs Recovery.

We follow the way in BERT [46] to mask an input SMILEs. First 15% tokens in a SMILEs will be randomly selected for masking and the minimum token number per SMILEs is one. For every selected token in a SMILEs, it has 85% chance to be changed to ¡MASK¿ token. With 10% and 5% chances, it will be randomly changed to any other token in the dictionary or keep unchanged correspondingly. The original SMILEs serve as ground truth for training the model but only the loss is only computed based on the outputs of masked tokens and the ground truths. By randomly masking the input SMILEs, the dataset used for pre-training model is enlarged. The randomness could increase the generalization ability of model and keep it from over-fitting.

The tokens are first embedded into the feature space. Besides the token embedding, positional-embedding is also included to add more sequence information used in self-attention layer to utilize the temporal information of the inputs.

The proposed SMILES-BERT differs from BERT in the following perspectives: 1) SMILES-BERT uses the single Masked SMILEs Recovery on large scale unlabeled dataset. 2) We do not include the segmentation embedding used in BERT into our model since we do not involve the continuous sentences training.

3.3.3 Fine-tuning for Molecular Property Prediction

The fine-tuning stage is shown in Figure 3.5. After pre-training on the large scale unlabeled SMILEs data, the model has a non-trivial initialization. During the pre-training, we pad every SMILEs with the leading token ¡GO¿. In the fine-tuning stage, the model output corresponding to the ¡GO¿ token is used for molecular property prediction.

A simple trainable classifier/regressor is added to the output of the ¡GO¿ token. Then the small scale of the labeled dataset is used for fine-tuning the model to predict specific molecular property.

The proposed SMILES-BERT has several advantages. First, it could use large scale unlabeled dataset for model pre-training. It not only contains the dataset itself, by randomly masking the inputs, but the dataset could also be enlarged into theoretically infinite. Second, unlike encoder-decoder structures in [30], the whole model involving in pre-training will be used in fine-tuning. Thus, the model could be more complicated since it does not need scaffolding parameters (the decoder parameters in Seq2seq model).

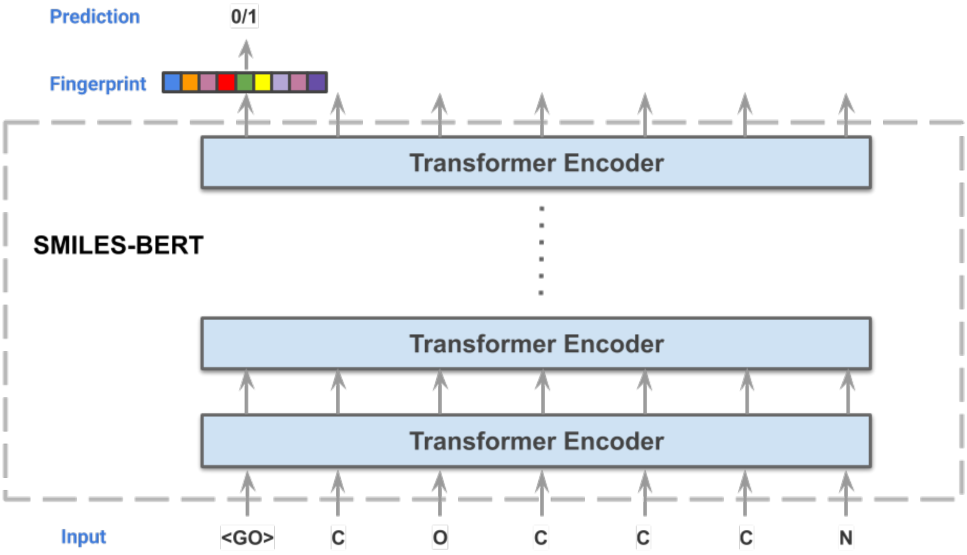


Figure 3.5. SMILES-BERT: fine-tuning stage..
smiles

3.3.4 Model Structure

In this paper, the proposed SMILES-BERT contains six Transformer Encoder layers. In each Transformer layer, the pre-attention and the post-attention fully-connected layers embed input features into a feature space with size 1024. For the attention block, SMILES-BERT uses a four-head multi-attention mechanism. Note that the layers and number of attention heads are less than the base BERT [46], which consists of twelve Transformer Encoder layers with 3072 fully-connected embedding size and twelve attention heads in attention block. It is because SMILEs are relatively simple than the natural language sequences. Besides, the vocabulary of SMILEs is much less than the vocabulary of natural language. We have tried the base structure setting of BERT to molecular properties prediction and it does not provide a noticeable improvement. Then we keep the SMILES-BERT in the current setting since it is better for the model to have less computation and memory requirements in practice.

3.4 Experiments

In this section, we describe all our experiments related details. First, the implementation details are given. Then we include the detailed settings in both pre-training and fine-tuning stages. Following that is a brief introduction to the datasets we include in our experiments. At last, we list the state-of-the-art methods used in our comparison and demonstrate the power of the proposed SMILES-BERT with a thorough discussion of the experimental results.

3.4.1 Implementation Details

The proposed SMILES-BERT is implemented with the FairSeq [102], which is Facebook AI Research Sequence-to-Sequence Toolkit written in Python and PyTorch. Along with the proposed SMILES-BERT, we also implement a series of fingerprint models based on modern natural language sequence learning models including RNNs-based models [32, 103, 104] and CNNs-based models [98, 99, 26, 100, 101] models. We plan to open source our implementations as well as our pre-trained models in the near future.

3.4.2 Experimental Settings

3.4.2.1 **Pre-training**

During the unsupervised pre-training stage, SMILEs are tokenized into tokens as the feeding inputs to SMILES-BERT. As the Masked SMILEs Recovery stage, the tokens are randomly selected to be masked with the masking strategy as described in Section 3.2. Note that the minimal number of masked token is set as one. Thus, each of the input SMILEs contains at least one masked token. In this way, the pre-training dataset is enlarged with randomness. With training on such dataset, the generalization capability of proposed SMILES-BERT is enhanced.

The pre-trained dataset we use for SMILEs is ZINC [105]. Zinc is a free database of commercially-available compounds for virtual screening. ZINC contains over 35 million purchasable compounds in ready-to-dock, 3D formats. ZINC is provided by the Irwin and Shoichet Laboratories in the Department of Pharmaceutical Chemistry at the University of California, San Francisco (UCSF). In SMILES-BERT, we only use the SMILEs of the molecules with no additional label to pre-train the SMILES-BERT, strengthening the model prediction capability than only using the labeled dataset. To verify the pre-train model, we randomly keep 10000 samples for validation and another 10000 for evaluation. The number in the training set ends up to 18,671,355.

We use Adam optimizer [106] as the pre-training optimizer. To better initialize the proposed model, a warm-up strategy is introduced for the first 4000 training steps. During the warm-up, the learning rate increases from 10^{-9} to 10^{-4}. We notice that the warm-up stage is crucial in SMILES-BERT pre-training. Without it, the model tends not to converge even after a long time training. After the warm-up finishes, the learning rate starts from 10^{-4} with the inversed-square-root updating strategy. The Adam betas are $(0.9, 0.999)$ and the weight decay is 0.1. The batch size is set to 256 and the dropout is set to 0.1.

We pre-train SMILES-BERT for 10 epochs on ZINC dataset. We use the exact recovery rate to evaluate the pre-train model. The exact recovery rate on the ZINC validation dataset is 82.85%, meaning 82.85% masked SMILEs could be exactly recovered by the information from the unmasked part.

Table 3.1. Parameters and Performances Contrast between Two Structures of SMILES-BERT

	layers	att heads	ffn dimension	dropout	accuracy (LogP)
SMILES-BERT	6	4	1024	0.1	**0.9154**
SMILES-BERT (large)	12	12	3072	0.5	0.9147

3.4.2.2 **Fine-tuning**

The supervised fine-tuning stage is based on the pre-trained model. As the pre-training stage, we use Adam optimizer for fine-tuning. The learning rate is not sensitive. We have tried several learning rates such as 10^{-5}, 10^{-6}, 10^{-7} and all the learning rate could get very good prediction results. Besides, we also test several different learning rate updating strategies such as no-updating, inversed-square-root updating. It turns out the updating strategy is not important for the training results. Thus, we simply choose not to update the learning rate in the fine-tuning stage.

In all our experiments, we fine-tune the model with each of the labeled datasets for 50 epochs and we choose the best model on validation data for the final evaluation.

3.4.3 Datasets Description

To evaluate our methods we use three datasets, LogP dataset, PM2 dataset and PCBA-686978 dataset in our experiments. The three datasets vary in not only properties but also the size of datasets. We would like to see if the pre-trained model could adapt well to fine-tuning with different molecular properties and different dataset sizes. The intrinsic logic of the experimental settings is from small-scale dataset (LogP) to large-scale datasets (PM2 and PCBA), from nonpublic datasets (LogP and PM2) to public dataset (PCBA).

3.4.3.1 **LogP**

LogP dataset is obtained from the National Center for Advancing Translational Sciences (NCATS) at National Institutes of Health (NIH). LogP dataset contains a total of 10,850 samples. Each sample contains a pair of a SMILEs string and a water-octanol partition coefficient (LogP) value. The value is continuous and we use the threshold of 1.88 suggested by an NCATS expert to convert the dataset as a classification task. Samples with LogP value larger than 1.88 will be classified as the positive samples, while the opposites are considered the negative ones.

3.4.3.2 **PM2**

PM2 dataset is also obtained from NCATS at NIH. PM2 has 323,242 data samples with PM2 labels. Similarly, the continuous PM2 labels are set as positive if it is larger than 0.024896; otherwise as negative.

3.4.3.3 **PCBA-686978**

PCBA [107] is a group of public available dataset containing 128 datasets from PubChem [108]. We select one of the largest datasets, the dataset with ID 686978 among the 128 datasets to evaluate our method. PCBA-93 contains 302,175 samples.

For each of the three datasets, we randomly select 80% for training, 10% as the validation set and the rest 10% for evaluation.

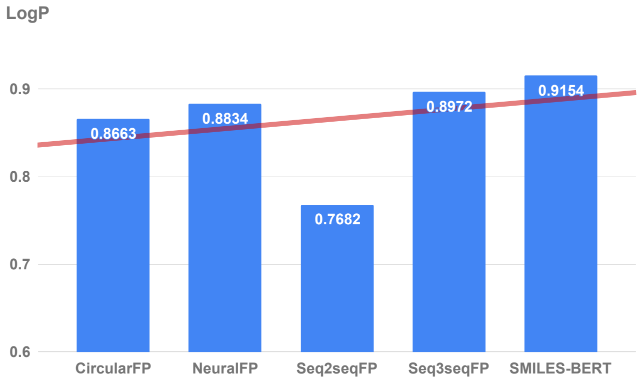
3.4.4 Experimental Results

3.4.4.1 **SMILES-BERT Structure Study**

To compare what kind of structure of SMILES-BERT could have better performance on molecular properties prediction tasks, we compare two structures. We have not explored more structures for the following two reasons. 1) Any of the two structures has better prediction performance (accuracy) than state-of-the-art method but they do not have a noticeable performance difference. 2) SMILES-BERT training could take a long time. For a single GPU, it could take more than a week to train the model for 10 epochs. The detailed parameters of the two structures are listed in Table 3.1 as well as the performance on LogP dataset.

In Table 3.1, the ffn dimension stands for the dimension for the shared fully-connected layer in each Transformer Layer. As shown in Table 3.1, the SMILES-BERT(large) is much more complicated than SMILES-BERT in all settings, while the performance is slightly worse. The performance difference could be caused by noise or randomness. However, we choose SMILES-BERT as our structure since it takes much less training cost and could have a very good performance.

Figure 3.6. Prediction Results (Accuracy) on LogP Dataset.



3.4.4.2 Comparison Methods

To prove the capability of molecular properties prediction performance of the proposed SMILES-BERT, we choose four state-of-the-art methods [30, 32, 22, 39] for comparison. These methods include one state-of-the-art manually designed fingerprint Circular Fingerprint [22], one graph-based neural network Neural Fingerprint [39], one unsupervised RNNs-based deep learning model Seq2seq Fingerprint [30], and one semi-supervised RNNs-based model [32]. We note the four methods as CircularFP, NeuralFP, Seq2seqFP, Seq3seqFP in all the following tables and figures.

3.4.4.3 **Results of LogP**

The prediction results for LogP data are shown in Figure 3.6. In this experiment, we use classification accuracy as the prediction metric to evaluate our model. As an unsupervised fingerprint, Seq2seq has reasonably lower performance than other methods. As a graph-based neural network, NeuralFP is slightly better than the manually designed CircularFP. Seq3seqFP and the proposed SMILES-BERT are both semi-supervised methods, which utilize large-scale unlabeled data. These semi-supervised methods have better performance than others. The proposed SMILES-BERT improves accuracy by around 2%. Since both of the SMILES-BERT and Seq3seq are pre-trained on Zinc, it shows SMILES-BERT could better utilize the unsupervised information with the Masked SMILEs Recovery task.

3.4.4.4 **Results of PM2**

PM2 is a much larger dataset than LogP. It contains 300 times data than LogP. It favors the supervised learning method because they could get better performance from more data samples. As shown in Table 3.2, unsupervised Seq2seqFP could not generate label-related fingerprint to have good prediction. The results of CircularFP and NeuralFP are similar. That CircularFP is slightly better than NeuralFP could be caused by that the graph-based neural network tends hard to train and tune in practice. Seq3seqFP slightly improves the performance compared to supervised method. The proposed SMILES-BERT achieves the better accuracy and it could get more than 5% improvement than Seq3seqFP. The results in Table 3.2 show that with the help of unsupervised pre-training, the proposed SMILES-BERT could have better representation and prediction capability after fine-tuning on the large dataset.

Table 3.2. Prediction Results (Accuracy on PM2 Dataset)

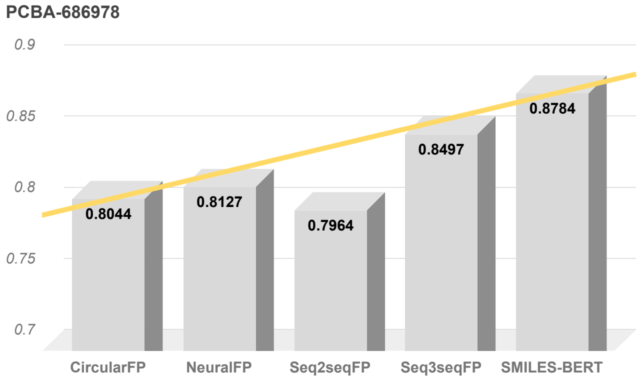
Method	Accuracy
Circular Fingerprint [22]	0.6858
Neural Fingerprint [39]	0.6802
Seq2seq Fingerprint [30]	0.6112
Seq3seq Fingerprint [32]	0.7038
SMILES-BERT	**0.7589**

3.4.4.5 **Results of PCBA-686978**

We introduce a public dataset PCBA-686978 to compare the molecular property prediction performance on all the state-of-the-art methods. Figure 3.7 shows the results of five models. The trend is the same as the LogP and PM2 datasets. The proposed SMILES-BERT has 87.84% accuracy, which is 8% higher than the unsupervised Seq2seqFP.

Figure 3.7. Prediction Results (Accuracy) on PCBA-686978 Dataset.

All the experiments on the three datasets demonstrate the power of the proposed SMILES-BERT. With the help of the large-scale unsupervised pre-training via the Masked SMILEs Recovery task, SMILES-BERT could easily be fine-tuning towards the labeled dataset. It could have outstanding molecular property prediction performance, independently from whether the scale of the labeled dataset is small or large.



3.5 Conclusions

In the thesis, to better use the numerous unlabeled molecular data and overcome some problems in current models, we have proposed a novel semi-supervised learning method SMILES-BERT for molecular properties prediction. The backbone of SMILES-BERT is BERT, a combination of Transformer Layer and attention mechanism. The semi-supervised method utilizes the power of unlabeled data through a large scale pre-training through a Masked SMILEs Recovery task. The labeled dataset could be easily fine-tuned on the pre-trained model and could have very good prediction performance. In our experiments on three datasets, i.e., LogP, PM2 and PCBA, the proposed SMILES-BERT over the performance of various of state-of-the-art methods and future potential to deal with most kind of label datasets with a good generalization capability.

In this work, we utilize the Masked SMILEs Recovery task in the pre-training stage corresponding to the masked language learning task in BERT [46]. However, BERT has another task to classify if two concatenated sentences are originally continuous. This task is to pre-train the classification with the input ¡GO¿ token. In SMILES-BERT, the classification capability of the model has not been involved in the pre-training stage. Thus, we could have the setting to include Quantitative Estimate of Druglikeness (QED) prediction as another task into the pre-training stage to warm up the classification capability of SMILES-BERT. It could potentially to increase the classification in the fine-tuning stage. We plan to design and include the QED prediction pre-training task in our future work.

CHAPTER 4

Conclusion

In the current era of big data, deep learning has been the state-of-the-art model for various applications. Image-based applications such as image classification, object detection, image segmentation, benefit most from deep learning networks. One reason for the successful applications of deep learning is that there are a large number of labeled training samples for the model to learn from. The cost of getting enough labeled data in health-related areas such as surgical video analysis and drug discovery is surprisingly high. It is significant to introduce unlabeled data during model training to improve the model performance.

In this thesis, we discuss two structured data, surgical video with sparsely label, and molecules in a sequential representation (SMILES). For surgical tool detection from surgical videos, we propose a series of approaches from regarding the problem as image classification to handling the problem with video classification in a semi-supervised learning way, using both spatial and temporal information, and both labeled and unlabeled frames. In the experiments, the proposed models not only demonstrate the superiority over state-of-the-art methods, but also show the necessity of using unlabeled data with temporal information. For molecule property prediction, we focus on the sequential representation since the sequential representation could be used to recover the structure of molecules. By introducing numerous unlabeled molecules for unsupervised pre-training, SMILES-BERT could learn the general information from molecules and the model could be easily adapted to specific molecular property prediction tasks.

We have demonstrated that in both surgical video analysis and molecule property prediction, introducing the unlabeled data could help improve the model performance. These proposed techniques could potentially be adapted into other health-related areas in which the cost of getting labeled data is still high.

REFERENCES

[1] D. Bouget, M. Allan, D. Stoyanov, and P. Jannin, "Vision-based and markerless surgical tool detection and tracking: a review of the literature," *Medical image analysis*, vol. 35, pp. 633–654, 2017.

[2] K. Cleary, H. Y. Chung, and S. K. Mun, "Or2020 workshop overview: operating room of the future," in *International Congress Series*, vol. 1268. Elsevier, 2004, pp. 847–852.

[3] F. Lalys and P. Jannin, "Surgical process modelling: a review," *International journal of computer assisted radiology and surgery*, vol. 9, no. 3, pp. 495–511, 2014.

[4] A. P. Twinanda, S. Shehata, D. Mutter, J. Marescaux, M. De Mathelin, and N. Padoy, "Endonet: a deep architecture for recognition tasks on laparoscopic videos," *IEEE transactions on medical imaging*, vol. 36, no. 1, pp. 86–97, 2017.

[5] S. Wang, Z. Xu, C. Yan, and J. Huang, "Graph convolutional nets for tool presence detection in surgical videos," in *International Conference on Information Processing in Medical Imaging.* Springer, 2019, pp. 467–478.

[6] Y. Li, C. Chen, X. Huang, and J. Huang, "Instrument tracking via online learning in retinal microsurgery," in *International Conference on Medical Image Computing and Computer-Assisted Intervention.* Springer, 2014, pp. 464–471.

[7] M. Ye, L. Zhang, S. Giannarou, and G.-Z. Yang, "Real-time 3d tracking of articulated tools for robotic surgery," in *International Conference on Medical Image Computing and Computer-Assisted Intervention.* Springer, 2016, pp. 386–394.

[8] L. Zhang, M. Ye, P.-L. Chan, and G.-Z. Yang, “Real-time surgical tool tracking and pose estimation using a hybrid cylindrical marker,” *International journal of computer assisted radiology and surgery*, vol. 12, no. 6, pp. 921–930, 2017.

[9] A. P. Twinanda, “Vision-based approaches for surgical activity recognition using laparoscopic and rbgd videos,” Ph.D. dissertation, Strasbourg, 2017.

[10] Y. Jin, H. Li, Q. Dou, H. Chen, J. Qin, C.-W. Fu, and P.-A. Heng, “Multi-task recurrent convolutional network with correlation loss for surgical video analysis,” *Medical image analysis*, vol. 59, p. 101572, 2020.

[11] H. Al Hajj, M. Lamard, P.-H. Conze, S. Roychowdhury, X. Hu, G. Maršalkaitė, O. Zisimopoulos, M. A. Dedmari, F. Zhao, J. Prellberg, *et al.*, “Cataracts: Challenge on automatic tool annotation for cataract surgery,” *Medical image analysis*, vol. 52, pp. 24–41, 2019.

[12] 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.

[13] O. Mohareri, J. Ischia, P. C. Black, C. Schneider, J. Lobo, L. Goldenberg, and S. E. Salcudean, “Intraoperative registered transrectal ultrasound guidance for robot-assisted laparoscopic radical prostatectomy,” *The Journal of urology*, vol. 193, no. 1, pp. 302–312, 2015.

[14] D. Bouget, R. Benenson, M. Omran, L. Riffaud, B. Schiele, and P. Jannin, “Detecting surgical tools by modelling local appearance and global shape,” *IEEE transactions on medical imaging*, vol. 34, no. 12, pp. 2603–2617, 2015.

[15] L. Zappella, B. Béjar, G. Hager, and R. Vidal, “Surgical gesture classification from video and kinematic data,” *Medical image analysis*, vol. 17, no. 7, pp. 732–745, 2013.

[16] D. Sarikaya, J. J. Corso, and K. A. Guru, "Detection and localization of robotic tools in robot-assisted surgery videos using deep neural networks for region proposal and detection," *IEEE transactions on medical imaging*, vol. 36, no. 7, pp. 1542–1549, 2017.

[17] R. Sznitman, C. Becker, and P. Fua, "Fast part-based classification for instrument detection in minimally invasive surgery," in *International Conference on Medical Image Computing and Computer-Assisted Intervention.* Springer, 2014, pp. 692–699.

[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[19] S. Wang, A. Raju, and J. Huang, "Deep learning based multi-label classification for surgical tool presence detection in laparoscopic videos," in *Biomedical Imaging (ISBI 2017), 2017 IEEE 14th International Symposium on.* IEEE, 2017, pp. 620–623.

[20] 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.

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

[22] 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.

[23] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks." in *CVPR*, vol. 1, no. 2, 2017, p. 3.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[25] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, “Residual attention network for image classification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3156–3164.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[27] 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.

[28] D. S. Kermany, M. Goldbaum, W. Cai, C. C. Valentim, H. Liang, S. L. Baxter, A. McKeown, G. Yang, X. Wu, F. Yan, *et al.*, “Identifying medical diagnoses and treatable diseases by image-based deep learning,” *Cell*, vol. 172, no. 5, pp. 1122–1131, 2018.

[29] 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.

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

[31] Y. Wang, J. Huang, W. Li, S. Wang, and C. Ding, “Specific and intrinsic sequence patterns extracted by deep learning from intra-protein binding and non-binding peptide fragments,” *Scientific reports*, vol. 7, no. 1, pp. 1–9, 2017.

[32] 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.

[33] S. Wang, Y. Guo, Y. Wang, H. Sun, and J. Huang, "Smiles-bert: Large scale unsupervised pre-training for molecular property prediction," in *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, 2019, pp. 429–436.

[34] X. Miao, X. Zhen, X. Liu, C. Deng, V. Athitsos, and H. Huang, "Direct shape regression networks for end-to-end face alignment," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5040–5049.

[35] X. Miao, X. Yuan, Y. Pu, and V. Athitsos, "λ-net: Reconstruct hyperspectral images from a snapshot measurement," in *IEEE/CVF Conference on Computer Vision (ICCV)*, vol. 1, 2019.

[36] X. Miao, X. Yuan, and P. Wilford, "Deep learning for compressive spectral imaging," in *Digital Holography and Three-Dimensional Imaging.* Optical Society of America, 2019, pp. M3B–3.

[37] F. Zheng, X. Miao, and H. Huang, "Fast vehicle identification via ranked semantic sampling based embedding," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence. AAAI Press*, 2018, pp. 3697–3703.

[38] L. Yue, X. Miao, P. Wang, B. Zhang, X. Zhen, and X. Cao, "Attentional alignment networks." in *BMVC*, vol. 2, no. 6, 2018, p. 7.

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

[40] K. Yang, K. Swanson, W. Jin, C. Coley, P. Eiden, H. Gao, A. Guzman-Perez, T. Hopper, B. Kelley, M. Mathea, *et al.*, "Are learned molecular representations ready for prime time?" *arXiv preprint arXiv:1904.01561*, 2019.

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

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

[43] Z. Xu, S. Wang, F. Zhu, and J. Huang, "Seq2seq fingerprint: An unsupervised deep molecular embedding for drug discovery," in *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics.* ACM, 2017, pp. 285–294.

[44] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[45] 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.

[46] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[47] C. Loukas, "Video content analysis of surgical procedures," *Surgical endoscopy*, vol. 32, no. 2, pp. 553–568, 2018.

[48] B. Choi, K. Jo, S. Choi, and J. Choi, "Surgical-tools detection based on convolutional neural network in laparoscopic robot-assisted surgery," in *Engineering in Medicine and Biology Society (EMBC), 2017 39th Annual International Conference of the IEEE.* IEEE, 2017, pp. 1756–1759.

[49] X. Hu, L. Yu, H. Chen, J. Qin, and P.-A. Heng, “Agnet: Attention-guided network for surgical tool presence detection,” in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support.* Springer, 2017, pp. 186–194.

[50] A. Jin, S. Yeung, J. Jopling, J. Krause, D. Azagury, A. Milstein, and L. Fei-Fei, “Tool detection and operative skill assessment in surgical videos using region-based convolutional neural networks,” in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV).* IEEE, 2018, pp. 691–699.

[51] M. Allan, S. Ourselin, S. Thompson, D. J. Hawkes, J. Kelly, and D. Stoyanov, “Toward detection and localization of instruments in minimally invasive surgery,” *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 4, pp. 1050–1058, 2012.

[52] M. Alsheakhali, M. Yigitsoy, A. Eslami, and N. Navab, “Surgical tool detection and tracking in retinal microsurgery,” in *Medical Imaging 2015: Image-Guided Procedures, Robotic Interventions, and Modeling*, vol. 9415. International Society for Optics and Photonics, 2015, p. 941511.

[53] S. Kumar, M. S. Narayanan, P. Singhal, J. J. Corso, and V. Krovi, “Product of tracking experts for visual tracking of surgical tools,” in *2013 IEEE International Conference on Automation Science and Engineering (CASE).* IEEE, 2013, pp. 480–485.

[54] N. Rieke, D. J. Tan, M. Alsheakhali, F. Tombari, C. A. di San Filippo, V. Belagiannis, A. Eslami, and N. Navab, “Surgical tool tracking and pose estimation in retinal microsurgery,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention.* Springer, 2015, pp. 266–273.

[55] R. Sznitman, R. Richa, R. H. Taylor, B. Jedynak, and G. D. Hager, “Unified detection and tracking of instruments during retinal microsurgery,” *IEEE*

transactions on pattern analysis and machine intelligence, vol. 35, no. 5, pp. 1263–1273, 2012.

[56] J. Zhou and S. Payandeh, "Visual tracking of laparoscopic instruments," *Journal of Automation and Control Engineering Vol*, vol. 2, no. 3, 2014.

[57] A. Reiter and P. K. Allen, "An online learning approach to in-vivo tracking using synergistic features," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 3441–3446.

[58] A. Reiter, P. K. Allen, and T. Zhao, "Marker-less articulated surgical tool detection," in *Computer assisted radiology and surgery*, 2012.

[59] S. Haase, J. Wasza, T. Kilgus, and J. Hornegger, "Laparoscopic instrument localization using a 3-d time-of-flight/rgb endoscope," in *2013 IEEE Workshop on Applications of Computer Vision (WACV)*. IEEE, 2013, pp. 449–454.

[60] S. Speidel, G. Sudra, J. Senemaud, M. Drentschew, B. P. Müller-Stich, C. Gutt, and R. Dillmann, "Recognition of risk situations based on endoscopic instrument tracking and knowledge based situation modeling," in *Medical Imaging 2008: Visualization, Image-Guided Procedures, and Modeling*, vol. 6918. International Society for Optics and Photonics, 2008, p. 69180X.

[61] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[62] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[63] M. Sahu, A. Mukhopadhyay, A. Szengel, and S. Zachow, "Addressing multilabel imbalance problem of surgical tool detection using cnn," *International journal of computer assisted radiology and surgery*, vol. 12, no. 6, pp. 1013–1020, 2017.

[64] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” in *Advances in Neural Information Processing Systems*, 2016, pp. 3844–3852.

[65] R. Li, S. Wang, F. Zhu, and J. Huang, “Adaptive graph convolutional neural networks,” in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[66] X. Wang and A. Gupta, “Videos as space-time region graphs,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 399–417.

[67] O. Chapelle, B. Scholkopf, and A. Zien, “Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews],” *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 542–542, 2009.

[68] K. M. Adal, D. Sidibé, S. Ali, E. Chaum, T. P. Karnowski, and F. Mériaudeau, “Automated detection of microaneurysms using scale-adapted blob analysis and semi-supervised learning,” *Computer methods and programs in biomedicine*, vol. 114, no. 1, pp. 1–10, 2014.

[69] G. Langs, A. Hanbury, B. Menze, and H. Müller, “Visceral: towards large data in medical imaging—challenges and directions,” in *MICCAI international workshop on medical content-based retrieval for clinical decision support.* Springer, 2012, pp. 92–98.

[70] G. Yengera, D. Mutter, J. Marescaux, and N. Padoy, “Less is more: Surgical phase recognition with less annotations through self-supervised pre-training of cnn-lstm networks,” *arXiv preprint arXiv:1805.08569*, 2018.

[71] J. Carreira and A. Zisserman, “Quo vadis, action recognition? a new model and the kinetics dataset,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE, 2017, pp. 4724–4733.

[72] C. Gu, C. Sun, S. Vijayanarasimhan, C. Pantofaru, D. A. Ross, G. Toderici, Y. Li, S. Ricco, R. Sukthankar, C. Schmid, *et al.*, “Ava: A video

dataset of spatio-temporally localized atomic visual actions," *arXiv preprint arXiv:1705.08421*, 2017.

[73] H. Zhao, Z. Yan, H. Wang, L. Torresani, and A. Torralba, "Slac: A sparsely labeled dataset for action classification and localization," *arXiv preprint arXiv:1712.09374*, 2017.

[74] Y. Jin, Q. Dou, H. Chen, L. Yu, J. Qin, C.-W. Fu, and P.-A. Heng, "Sv-rcnet: Workflow recognition from surgical videos using recurrent convolutional network," *IEEE transactions on medical imaging*, vol. 37, no. 5, pp. 1114–1126, 2018.

[75] M.-L. Zhang and Z.-H. Zhou, "Multilabel neural networks with applications to functional genomics and text categorization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 10, pp. 1338–1351, 2006.

[76] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[77] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.

[78] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 807–814.

[79] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[80] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853*, 2015.

[81] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *Journal of Artificial Intelligence Research*, vol. 11, pp. 169–198, 1999.

[82] T. Durand, T. Mordan, N. Thome, and M. Cord, "Wildcat: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, vol. 2, 2017.

[83] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[84] C. Liu, B. Zoph, M. Neumann, J. Shlens, W. Hua, L.-J. Li, L. Fei-Fei, A. Yuille, J. Huang, and K. Murphy, "Progressive neural architecture search," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 19–34.

[85] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[86] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[87] 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.

[88] 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.

[89] 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.

[90] 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.

[91] 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.

[92] 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.

[93] 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.

[94] 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.

[95] 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.

[96] 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.

[97] 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.

[98] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 933–941.

[99] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1243–1252.

[100] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *arXiv preprint arXiv:1802.05365*, 2018.

[101] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf*, 2018.

[102] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, "fairseq: A fast, extensible toolkit for sequence modeling," *arXiv preprint arXiv:1904.01038*, 2019.

[103] S. Wiseman and A. M. Rush, "Sequence-to-sequence learning as beam-search optimization," *arXiv preprint arXiv:1606.02960*, 2016.

[104] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[105] 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.

[106] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[107] B. Ramsundar, S. Kearnes, P. Riley, D. Webster, D. Konerding, and V. Pande, "Massively multitask networks for drug discovery," *arXiv preprint arXiv:1502.02072*, 2015.

[108] Y. Wang, J. Xiao, T. O. Suzek, J. Zhang, J. Wang, Z. Zhou, L. Han, K. Karapetyan, S. Dracheva, B. A. Shoemaker, *et al.*, "Pubchem's bioassay database," *Nucleic acids research*, vol. 40, no. D1, pp. D400–D412, 2011.

BIOGRAPHICAL STATEMENT

Sheng Wang received his Ph.D. in Computer Science and Engineering from the University of Texas at Arlington in 2020. Prior to beginning the Ph.D. program, Sheng obtained his M.S. and B.S. degree from Sichuan University, China. His main research interests are deep learning and semi-supervised learning with medical imaging, and bioinformatics. During his Ph.D. study, 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), International conference on Information Processing in Medical Imaging (IPMI).