QATAR UNIVERSITY

COLLEGE OF ENGINEERING

DIFFERENTIAL ARCHITECTURE SEARCH IN DEEP LEARNING FOR DNA SPLICE SITE CLASSIFICATION

BY

SHABIR MOOSA

A Thesis Submitted to

the Faculty of the College of Engineering

in Partial Fulfillment of the Requirements for the Degree of

Masters of Science in Computing

June 2019

COMMITTEE PAGE

The members of the committee approve the Thesis of

Shabir Moosa defended on 17/04/2019.

Prof. Abbes Amira

Thesis Supervisor

Dr. Sabri Boughorbel

Thesis Co-Supervisor

Dr. Puthen Veettil Jithesh

Committee Member

Dr. Abdelkarim Erradi

Committee Member

Approved:

Abdel Magid Hamouda, Dean, Engineering

ABSTRACT

Moosa, Shabir, Masters:

June: 2019, Masters of Science in Computing

Title: Differential Architecture Search in Deep Learning for DNA Splice Site Classification

Supervisor of Thesis: Prof. Abbes Amira

The data explosion caused by unprecedented advancements in the field of genomics is constantly challenging the conventional methods used in the interpretation of the human genome. The demand for robust algorithms over the recent years has brought huge success in the field of Deep Learning (DL) in solving many difficult tasks in image, speech and natural language processing by automating the manual process of architecture design. This has been fueled through the development of new DL architectures. Yet genomics possesses unique challenges as we expect DL to provide a super human intelligence that easily interprets a human genome. In this thesis, the state-of-the art DL approach based on Differential Architecture Search (DARTS) mechanism was adapted for interpretation of biological sequences. This method has been applied to the splice site recognition task on raw Deoxyribonucleic Acid (DNA) sequences to discover high-performance convolutional architectures by automated engineering. The discovered architecture outperformed fixed DL architectures. As part of the study, the DARTS model was benchmarked on CPU and multiple GPU architectures in terms of computational time and classification performance. The results have shown a potential of using this automated architecture search mechanism for solving other problems in genomics.

DEDICATION

"For my family and friends, for their constant support, love and faith in me"

ACKNOWLEDGEMENTS

First and foremost, I thank Allah, the almighty, for giving me the strength, knowledge, ability and opportunity to undertake this research study and be able to complete it satisfactorily. Without his blessings, this achievement would not have been possible.

I would also like to express my heartfelt gratitude to my parents for their unconditional love, encouragement and prayers. I thank with love to my wife and daughter who are the biggest source of strength and happiness for me. My acknowledgement would be incomplete without thanking my friends for keeping me going on my path to success and for ensuring that good times keep flowing.

I would like to submit my sincere and deepest gratitude to my supervisors, Dr. Abbes Amira and Dr. Sabri Boughorbel for their meticulous supervision, admirable patience, and constant support. I am deeply indebted to their highly commendable efforts during my journey in research. I am also grateful to the team at Sidra Medicine, for their persistent support given to me at all times.

I would also like to thank all my professors and friends at Qatar University for their support, encouragement and credible ideas during the entire course of my Masters. Special thanks go to Zeineb, Hisham, George, Hakem and Saad.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CNN Convolutional Neural Networks

DA Deep Autoencoder

DARTS Differential Architecture Search

DBM Deep Boltzmann Machine

DBN Deep Belief Network

DL Deep Learning

DNA Deoxyribonucleic Acid

DNN Deep Neural Network

GPU Graphical Processing Unit

GRU Gated Recurrent Units

LSTM Long Short Term Memory

ML Machine Learning

NAO Neural Architecture Optimization

NAS Neural Architecture Search

NGS Next Generation Sequencing

PPM Path Towards Personalized Medicine

QGP Qatar Genome Program

QNRF Qatar National Research Fund

RNN Recurrent Neural Network

SGD Stochastic Gradient Descent

TPU Tensor Processing Unit

CHAPTER 1: INTRODUCTION

Human Genome

The Human Genome consists of a set of nucleic acid sequences encoded as DNA molecule. The DNA forms a double helix shape and is composed of nucleotides. Each nucleotide is composed of a nucleoside and a phosphate group. A nucleoside is a nitrogenous base and a sugar. The nitrogenous bases are of 4 types; Adenine(T), Thymine(T), Guanine(G) and Cytosine(C). The information in a DNA is stored in these four chemical bases. As shown in Figure 1.1, the bases pair up with each other in such a way that A pairs with T and G pairs with C. Almost 99 percent of the bases are same in all humans and their order provides information for building and maintaining the human body. Each DNA strand forms the shape of a spiral ladder with the base pairs attached to the sugar-phosphate backbone.

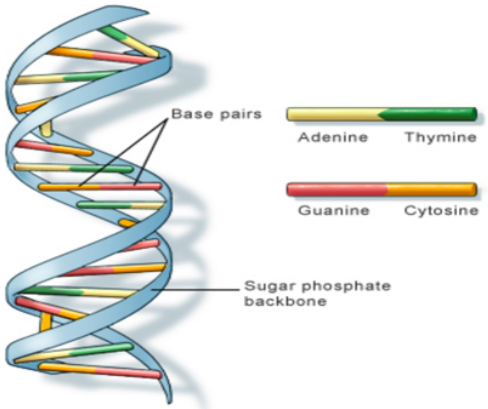


Figure 1.1: An illustration of DNA.

The prosperity of research in genomics started ever since the DNA molecules were interpreted by Watson and Crick [1] as the medium of genetic information. Researchers have been ever since striving to decipher the biological process using this genetic information in order to study the roles assumed by multiple genetics factors under different environmental conditions. Genomic research has become data

intensive with the advancement of Next Generation Sequencing (NGS) techniques with the capability of sequencing an entire human genome in a single day. The vast amount of information generated can be used for conducting scientific study using statistical methods. These methods can be used to recognize patterns in the DNA sequences by identifying various genomic elements like introns, exons, positioned nucleosomes, promoters, enhancers and splice site junctions.

Genomics Research in Qatar

Recently, there has been an increased interest in genomics research in Qatar which could lead to great advancement in the field of precision medicine and personalized healthcare. This section highlights the areas of genomic research initiatives in Qatar. The Qatar Genome Program (QGP), launched in December 2013, is a national initiative that aims to map the genome of Qatari population. The aim of the project is to develop personalized healthcare by incorporating advanced and innovative genomic technologies into research and medicine. Individual research studies have further contributed to the availability of whole genomes and exomes providing insights to the genetic structure of the Qatari population [2]. The Qatar National Research Fund (QNRF) along with the collaboration of QGP launched the initiative of Path Towards Personalized Medicine (PPM) in 2015 to support and advance research activities focused on providing personalized medical treatment tailored to patient's individual characteristics.

Motivation

The genomics community is rapidly embracing DL techniques to solve many of the problems in the field. But DL is yet to revolutionize the genomics domain to resolve many of the pressing challenges [3]. The intersection of genomic research and current state-of-the-art DL methods in computer vision, speech recognition and natural

language processing will lead to an insightful understanding of genomics that will benefit multiple fields like precision medicine and drug discovery. Hence, there is a need to adopt the powerful DL methods utilized in other domains to tackle the challenges in the field of genomics. The research work is carried out at SIDRA and the methods will be evaluated on SIDRA's High Performance Computing platform.

Problem Statement

Deep Learning is shifting the paradigm in Machine Learning (ML) from feature engineering and algorithmic development into a single framework where no hand crafted feature are needed. However, the recent trends in DL have shown that much effort is dedicated to the design of complex neural network architectures. For example in ImageNet competition, through 4 years the winning model went from 8 layers [4] in 2012 to 152 layers [5] in 2016 with various choices on architectures parameters such as skip connections, filter sizes, repeated blocks etc. The problem lifted by DL of feature engineering is creating an additional burden of architecture engineering. Therefore there is a need to automate the process of manual architecture definition to a data-driven approach. Our objective is to explore some recent DL techniques to tackle this issue in genomics domain. Many of the architectures used in genomics have been borrowed from computer vision and natural language processing domains and still undergo the manual tasks of feature and architecture engineering. To the best of our knowledge this is the first attempt to apply and adapt similar techniques for genomics data.

Thesis Objective

The focus of this thesis is to adapt the DARTS technique in [6] for splice site classification in genomics domain. The study was performed on the Splice Site Recognition (SSR) dataset which provides a scope for analysis of the human genome and iden-

tification of unknown regions to understand the biochemical processes involved in building and maintaining a human body. The following are the objectives of the thesis:

- Conduct a literature review on existing DL approaches performed in solving problems in the field of genomics.
- Develop new DL convolutional architectures using DARTS approach for splice site classification.
- Evaluate the performance of the discovered architecture against fixed architectures based on Recurrent Neural Network (RNN) and Convolutional Neural Networks (CNN).
- Benchmark the architecture performance over CPU and multiple GPU models in terms of execution time, learning and inference speed.

Research Contributions

The main contributions of this thesis are shown in the following.

- A literature review is conducted on the application of DL techniques in genomics. For each of the selected studies, the addressed problem, DL architecture used , dataset, and the achieved performance are presented. The methods used to address challenges were discussed, and a comparison between the reviewed studies is presented.
- A recently proposed continuous architecture search method, DARTS, was adapted and applied on DNA data. This was initially implemented on image dataset. This is the first research study that has applied Neural Architecture Search (NAS) to solve a problem in genomics domain.

- A data loader for DNA sequence was written as the model input initially was images. This was adapted to take sequence data represented by a four-dimensional hot-vectors representing the four base pairs (A,C, T,G).
- For model evaluation, the original code written in pytorch worked only for Graphical Processing Unit (GPU). All the necessary adaptation to handle CPU processing were introduced.
- For computational performance analysis, the precision of model weights was changed to experiment with different precision levels of single, half and double floating point operations. The learning and evaluation experiments adapted to benchmark on the three data types in the each of the forward and backward passes.
- Implemented visualization graphs for the experiments and interactive standard and mean deviation plots of the results.

Thesis Outline

The thesis is organized as follows. In Chapter 2, we present the necessary background information of the existing approaches for the splice site classification problem. In Chapter 3, the DARTS algorithm is discussed in detail. Our adapted approach and the materials and methods used are discussed in Chapter 4. Chapter 5 presents an evaluation of the results obtained. Finally, Chapter 6 concludes the thesis and proposes some future work.

CHAPTER 2: BACKGROUND AND LITERATURE REVIEW

We present the background needed for the remainder of the thesis in this chapter. In Section 2.1, we present an overview of DL models. Section 2.3 presents the advancements in the field of DL. Section 2.4 introduces the splice site recognition problem. Section 2.5 presents the literature review of the existing methods used for prediction and classification of splice sites. The literature was specifically searched for studies that applied ML and DL techniques and taking into review the various architectures used. In Section 2.6, we present a table with a comparative analysis of the studies

Overview of Deep Learning

Deep Learning is a class of ML algorithms that combines raw inputs into layers of intermediate features. They take raw features from large datasets and use them to create a predictive tool from the patterns hidden inside the data. They have shown impressive results over existing best-in class ML algorithms across various domains. For the past five years, DL algorithms have revolutionized fields such as high-energy physics [7], computational chemistry [8] and dermatology [9]. The off-the-shelf implementation of these algorithms across different fields have produced comparable or higher accuracies than previous state-of-the art methods that required extensive customization over year.

Deep Learning techniques rely on artificial neural networks which grew from research on artificial neurons that are similar to the neurons in the brain. The history of artificial neural networks dates back to first computational model proposed in [10] where layers of neuron-like nodes mimic how neurons in the brain process information. The neural networks are composed of an input layer, one or more hidden layers and eventually a link to the output layer as shown in Figure 2.1. A layer has a set of nodes which are connected using edges to the immediate earlier and deeper lay-

ers. The nodes in certain neural networks connect to itself with a delay. The input layers consist of the variables measured in the data set of interest. There are multiple hidden layers in neural networks where each layer does feature construction for the layers before it. The increasing levels of abstraction allows efficient learning of data representation. The higher level abstractions are learned by representing them at lower levels as less abstraction. These are very powerful techniques as they help to learn from the raw data directly [11] and can be used in many fields. DL has been successful in areas such as computer vision, image recognition, speech translation, object detection and robotics and is most useful when large datasets are available.

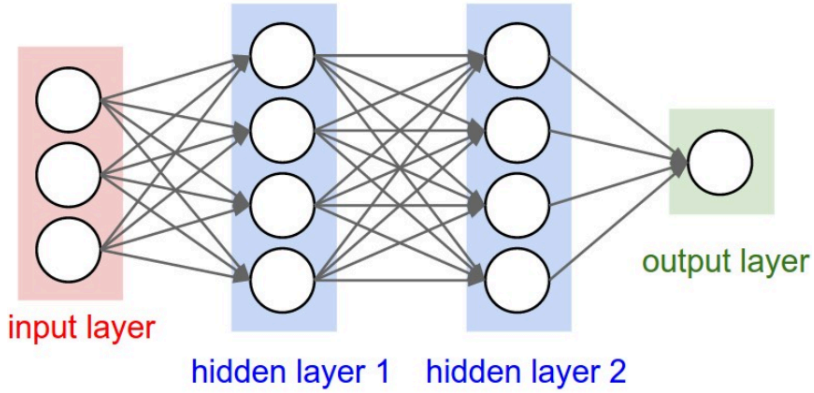


Figure 2.1: DL architecture: Multilayer perceptron.

The intersection of DL and genomic research has led to insightful understanding of the genomics domain. Although DL is still in early stages in genomics, it has influenced many fields such as cancer diagnosis and treatment, population genetics and functional genomics. It has been particularly useful in gene expression and disease analysis [12] [13] for predicting the effects of mutations in non-coding regions in a DNA. It has made effective contributions in the detection of skin cancer [9] and breast cancer [14]. Several DL architectures as shown in Fig. 2.2 have been used in the field of genomics such as RNN, CNN, Deep Neural Network (DNN), Deep Belief Network (DBN), Deep Boltzmann Machine (DBM) and Deep Autoencoder (DA).

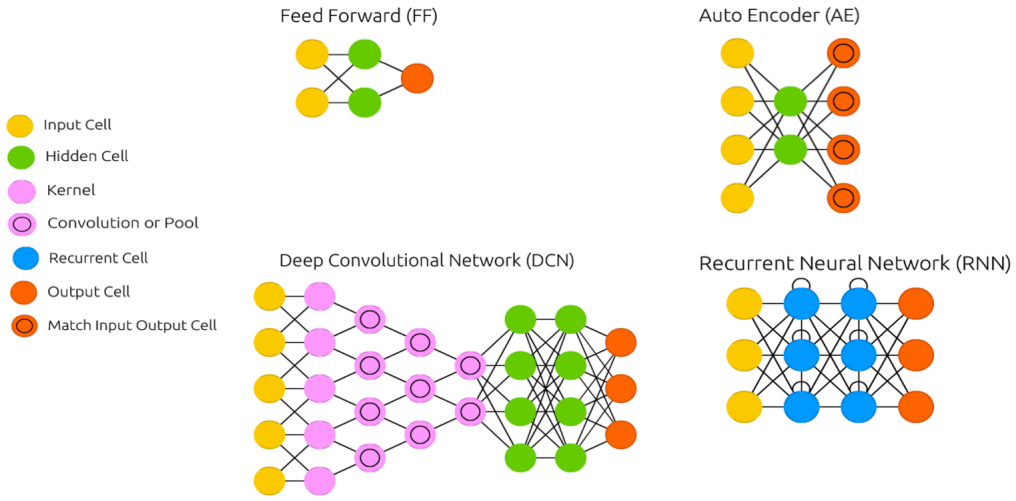


Figure 2.2: Types of neural networks.

These different types of architectures have been used to solve various specific problems in genomics. For example, the CNN architectures have been used for problems in image classification tasks and have been widely adopted in genomics for automatically learning the characteristics of gene data. The RNNs used in speech recognition problems have found to be efficient in handling DNA sequence data. The following section provides a review of how various DL architectures have been used in the perspective of genomics study.

Convolutional Neural Networks

CNN have been most successful in the field of image processing and computer vision due to their ability to examine spatial information. In genomics, the fundamental building blocks of CNN used in computer vision [4] was utilized for feature extraction by representation of the DNA sequence as an image. CNN architectures have been used for discovering sequence patterns of genomic sequence motifs to perform classification tasks [15]. Several three layer CNN models such as DeepSEA [16] have been developed for sequence based prediction and classification problems and a similar architecture in [17] to analyze the functional activities in a genome sequence.

CNN models [18] and DeepBind [19] were proposed to understand sequence specificity involved in binding of proteins.

Though CNN architectures have shown superior performance over existing methods, poor architecture design would result in inefficient performance of these tuned models [18]. Therefore, it is of utmost importance for the researcher to have biological background and an in-depth understanding of the CNN architecture to optimize and choose a particular architecture for a specific task. Increasing the depth of the network will not have much effect, but choice and design of convolutional and pooling operations, window and kernel size could result in much improved performance.

Recurrent Neural Networks

RNN have shown impressive results in the field of natural language processing, speech recognition and translation. They have proven to outperform CNN and other DL models on sequential data due to their memory like property to store information about past inputs. The inputs are scanned in a sequential manner and the output is a function of the the current input and previous hidden layer. A variance of the RNN is the bi-directional model proposed in [20] where the future inputs are considered as well. Subsequently to solve the problem of vanishing gradient Long Short Term Memory (LSTM) was proposed in [21] and later Gated Recurrent Units (GRU) in [22] that truncated the gradient propagation.

The sequential nature of the DNA made RNN models applicable to use in many scenarios. A neural machine translation model based on LSTM was proposed in that converted the protein prediction problem as similar to a language translation task. Many other models based on RNN such as DeepNano [23], a hybrid convolutional LSTM model [24] named DanQ were built for prediction from protein sequences. Recently proposed seq-to-seq model based on RNN [25] has been applied for prediction of secondary protein structure.

Autoencoders

Autoencoders were initially used for the initialization of network weights and have now been used for deeper architectures by stacking a number of autoencoders to be extended to form denoising autoencoder [26], contractive autoencoder [27], sparse autoencoder [28] and variational autoencoder [29]. They have been successful due to their ability to learn from compressed input representation through encode-decode method. Autoencoders have been used for dimension reduction in gene expression and also for gene clustering tasks. The autoencoders do not guarantee improvement in the model with higher reconstruction accuracy. Variational autoencoders are best suited for genomic tasks because of their ability to deal with complex dependencies that exists. A two step model was proposed in [30] for prediction in drug responses with an initial prediction by unsupervised model and then by a semi-supervised prediction model. The unsupervised model called the Perturbation variational autoencoder embedded the data space to a lower dimensional latent space to model the drug-induced effect into a linear function. The semi-supervised prediction model Drug Response variational autoencoder is then trained jointly to model both drug-induced effects as well as the outcome of the treatment. These variational autoencoder frameworks have led to improvement in predicting drug responses.

Architecture Designing in Deep Learning

Designing a DL architecture is one of the key hyperparameters to solve problems using DL techniques. At present, DL architectures are designed manually by human experts by handcrafting through careful experimentation or modifying existing networks. CNN architectures are a class of neural architectures that consist of alternating layers of convolution and pooling operations. The final layers consist of fully connected layers and the last layer is a softmax classifier as shown in Figure 2.3. These architectures are trained using back-propagation via Stochastic Gradient De-

scent (SGD) to determine the network weights and biases to minimize the loss function for mapping the input to outputs. The CNN has a set of learnable kernels or

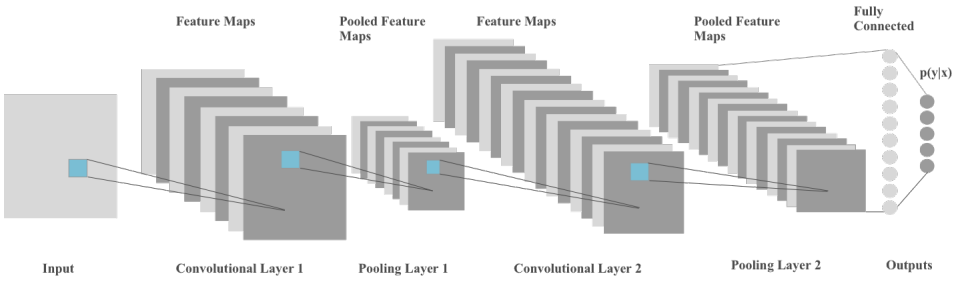


Figure 2.3: CNN Architecture.

features that extracts features from input data to produce feature maps. Feature maps are generated by sliding the filter over input and performing the dot product or the convolution operation, followed by a non-linear activation function such as sigmoid, ReLU, and tanh. The activation function is to introduce non-linearity in the model. ReLU is the preferred activation function as it speeds up training better than others [4]. The weights are shared among feature maps to reduce the number of parameters and also to detect the same feature irrespective of its location in the inputs [31]. The feature map size depends on the size of the filter and the stride, so when convolution on an input image of size (IXI) by a filter of size (FXF) with stride S, the output size will be (OXO) given by:

$$O = \frac{I - F}{S} + 1 \tag{2.1}$$

The pooling operation reduces the resolution of the feature maps from the previous convolution operation. The input is split into disjoint regions and an output of size (RXR) from each region is produced. There are two types of pooling operations: max pooling or average pooling [32]. When a feature map of size (OXO) is fed into pooling layer, the output is given by:

$$P = \frac{I}{R} \tag{2.2}$$

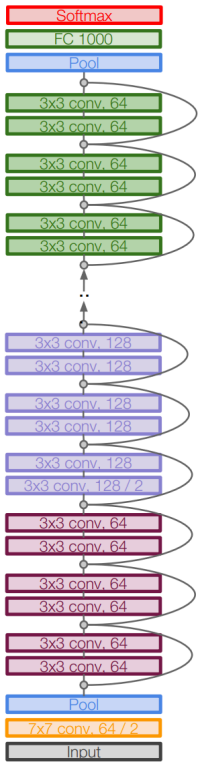
The final layer of CNN consists of a fully connected layer which extracts the global features from the inputs. This layer is similar to a feed-forward neural network where the units are connected to all the hidden units in the previous layer. The last layer is a softmax layer which calculates the probability over K given classes. The availability of more number of classes to predict from implies that more features need to be learned. This requires more filters increasing the total number of parameters in the network. There has been improvements in CNN architectures from the start of 1989 to date in optimization of parameters, regularization, structural changes in designing of new blocks, etc. Many CNN architectures have been introduced over the years that has increased the complexity of architecture and these designs have been based on intuition and engineering. The presence of a number of possible variations and choices makes the design space of CNN architectures extremely hard for a manual exhaustive search. A number of architectures were introduced over the years such as VGGNet [4] as shown in Fig. 2.4a which had used filters with a small receptive field of 3x3 and a convolution stride of 1. The spatial pooling had 5 layers of max pooling operations. The image is padded to preserve the spatial resolution after convolution. The hidden layers were equipped with ReLU operations. This architecture mainly addressed the depth aspect of the convnet architectures design.Another major CNN architecture introduced by google around 2014 in the ImageNet Large-Scale Visual Recognition Challenge was GoogleNet [33] shown in Fig. 2.4c. The simple convolution layer was replaced with an inception module which performed 3x3, 5x5 convolution operations with 3x3 max pooling and 1x1 convolution operation operating independently on the feature maps from the previous layers. All the feature maps were then concatenated in the end. The ResNet model [5] shown in Fig. 2.4b was proposed in 2015 which introduced a novel architecture with skip connections and batch normalization. The skip connections were gated recurrent units similar to

the one in RNN. The architecture had 152 layers and was able to outperform human level performance.

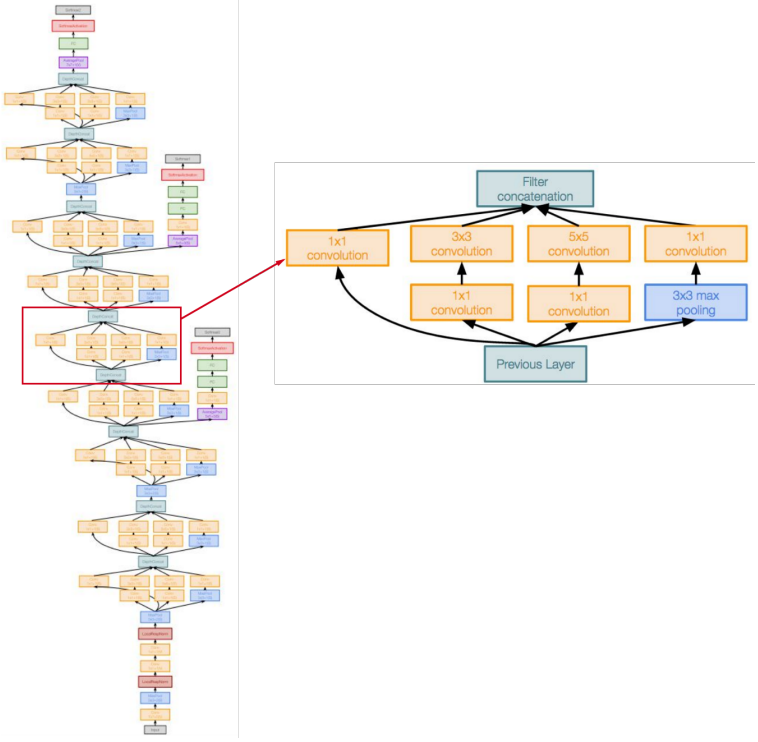
Advancement in Deep Learning

The advancement of neural networks have accomplished major achievements in the field of image classification, object detection and natural language processing. Designing these neural network architectures requires computational resources and significant efforts from human experts in DL through trial and error. Over the recent years, there has been a paradigm shift from feature designing to architecture designing in the field of image classification and natural language processing [34, 35, 36, 37] to develop algorithmic solutions for automating the manual process of architecture design using NAS methods. They have provided promising results in designing models better than human designed ones on some benchmark datasets. Architecture search aims to automate every aspect of ML so that non-experts can apply ML to solve their problems at hand. The goal is to find an optimal architecture from a given search space so that the validation accuracy on a particular task is maximized. NAS has some similarities to program synthesis and inductive programming where a program is searched from examples [38]. Many architecture search algorithms have been proposed such as Reinforcement Learning (RL) [37] which uses a policy gradient algorithm to optimize architecture configurations. This approach is computationally expensive and time consuming as they design and train each network from scratch during the exploration in the search space. Several approaches were proposed for improving the efficiency of NAS such as establishing a particular structure for the search space [34], performance prediction [39] and weight prediction of individual architectures [40] and by a parameter sharing mechanism [41] across multiple architectures. A novel approach of searching the architectures over a continuous domain alternate to searching over a discrete set of child architectures were proposed

(a) VGGNet Architecture.



(b) ResNet Architecture.



(c) GoogleNet Architecture.

Figure 2.4: CNN Architectures.

by DARTS [6] and Neural Architecture Optimization (NAO) [42]. These continuous search mechanisms has outperformed other architecture search algorithms by achieving competitive performance over a rich architecture search space and using less computational resources. Both DARTS and NAO performs continuous optimization using gradient based method to find the best architecture. The difference in both approaches lies in the optimization space and method of deriving the best architecture from the continuous space. The NAO uses a decoder network to derive the architecture whereas in the DARTS approach the discrete architecture is obtained by $argmax$ of the mixture of operations in the network.

Splice Site Recognition Problem

Proteins form an essential component in all living organisms and a major biological process that occurs in all living cells is the production of proteins. They play a vital role in the biochemical reactions within cells and in metabolism. The protein coding process called gene expression occurs in two stages: Transcription and Translation. During Transcription, the DNA is copied to produce an mRNA (messenger Ribonucleic Acid). The protein coding process occurs in the translation phase where the mRNA sequence is decoded to produce the proteins.

Eukaryotic organisms are complex organisms with cells in which the genetic material is made up of a membrane-bound nucleus or nuclei. Their genes are composed of alternated segments of introns and exons. Exons form the coding regions in a DNA during translation to proteins. The biological significance of intronic regions are not known yet as they do not participate in the protein building process. During translation stage in eukaryotes, the process of splicing occurs where the introns are spliced out from the mRNA molecule. Splice site classification is a sub-problem of gene prediction to correctly recognize genes from a DNA sequence. This prediction is hard due to the mechanism of splicing to remove the introns and combine exons.

The exons are short in length and has few nucleotides, whereas introns are longer. The number of introns and their length vary among species and even within the same species. The introns vary differently and the ones involved in splicing are called spliceosomal introns. The spliceosomal introns follow the GT-AG rule where the 5' end begins with nucleotides GT and end with AG at the 3' end. These borders are referred to as splice sites. The boundary points where splicing occurs on a gene sequence are called splice junctions or splice sites as shown in Fig. 2.5, where the GT is called the donor and AG is called the acceptor splice site.

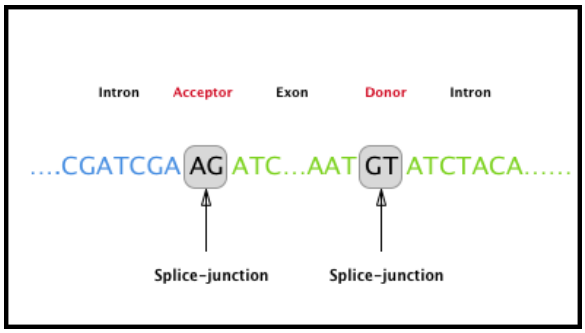


Figure 2.5: Splice-junction sites.

The problem of splice site recognition can be formulated by the following: Given an unclassified DNA sequence, classify it as one having a splice site or a non-splice site sequence. An efficient learning algorithm is required where the objective is to determine a classification rule that can accurately identify a region or pattern to distinguish a genomic DNA sequence.

Splice Site Detection Methods

Precise identification of Exon-Intron(EI) junctions or donors and Intron-Exon junctions(IE) or acceptors from a sequence is beneficial for advancements in transcriptome research and is a crucial step for fully understanding the gene expression. The accurate detection of splice junctions is challenging because of the high rate of false positives caused by the presence of short canonical splicing patterns [43].

There are currently two different techniques used to solve the splice junction prediction problem: Alignment based techniques and ML based. The sequence alignment-based techniques map millions of short RNA sequences produced by RNA-seq to the reference genome and then estimate where splicing occurs by identifying the adjacent exon locations. The existing alignment-based techniques such as SpliceMap [44] and TopHat [45] detect only canonical splice sites while missing the non-canonical sites which are required for accurate junction prediction. The ML based techniques can predict non-canonical sites as well by appropriate training. Different ML approaches have been used such as Support Vector Machines(SVM) [46, 47, 48, 49] Random Forest(RF) [50], Decision Trees (DT) [51], Naïve Bayesian(NB) [52], Markov Model [53] and AdaBoost [54] to identify splice or non-splice sites. Among them SVM models have been used very often due to their capability to handle high-dimensional datasets. However, certain kernel and penalty parameters in SVM require extensive tuning which is time consuming. The effectiveness of all these approaches also depends on the feature engineering technique used, which is a major initial step in solving a classification problem. Many feature engineering techniques have been proposed for feature construction directly from the DNA sequence, such as the MM1 (1-order Markov model) in [46] for feature construction from splice site sequences and using the SVM for prediction. The features were constructed based on different statistical approach [50] [47] with automated feature extraction proposed in [52] for prediction of splice sites. A length-variable Markov Model (LVMM) which produced higher accuracy with low time cost was discussed by [53]. A hybrid algorithm of AdaBoost classifier was proposed in [54] which provided an improvement in performance compared to the other approaches. The efficacy of all these approaches is based on the feature extraction step which is often a tedious task that is performed by domain experts. Manual operation of feature extraction often leads to incomplete representation or one-high dimensional feature space which will cause problems in

the ML process. The challenges involved in performing manual feature extraction and model training has led to a demand for DL based computational methods that performed automated feature representation. Many DL architectures were used and developed for the splice site prediction based on CNN [55, 56, 57, 58, 59, 60], RNN [43, 61, 62, 63, 64, 65, 66], Restricted Boltzmann Machines (RBM) [67], Autoencoders [63, 68] and Deep Belief Networks [67]. The steady performance of DL architectures has led to the design of many complex hybrid architectures [69, 70]. Although these DL architectures have removed the burden of manual feature extraction, they are still time consuming due to the manual effort present in designing efficient architectures. In general, the existing methods still utilise the manual effort in designing architectures which needs a lot of expert domain knowledge and is time consuming.

Comparative Analysis

In this section, the comparison of the reviewed studies is presented in Table 2.1. The criteria for comparison are the following:

1. The referenced study.
2. Datasets used in the study.
3. Type of DL architecture used.
4. Method of DNA representation used: One-hot Encoding, Word Embeddings, Orthogonal Encoding, Label Encoder, Orthonormal Sparse Encoding.
5. Metrics used for evaluation of the proposed approach.
6. Summary of the study.

Table 2.1: Comparative Analysis of Literature Review

Reference	Dataset used	DL method	Input representation	Reported classification metrics	Summary
[43]	UCSC database	RNN	One-hot Encoding	Accuracy	Proposed and tested RNN architectures to model DNA sequences and to detect splice junctions.
[55]	Splice Site Recognition (SSR) dataset	CNN	Label Encoder	Sensitivity, Specificity, Accuracy, AUC	Proposed a shallow version of ResNet CNN architecture for splice site classification.
[57]	Gencode database	Deep CNN	Label Encoder	Accuracy, Recall, Precision, F1-score	Proposed a deep CNN based appproach for prediction of splice junctions.
[58]	Histone, Splice, Promoter	CNN	One-hot Encoding	Accuracy	Proposed a new approach for classifying DNA sequences using CNN by representing DNAsequences as text data.
[59]	HS3D and GENCODE datasets	CNN	Orthonormal Sparse Encoding	Accuracy, Sensitivity, Specificity	Presented a deep learning based splice junction sequence classifier, named DeepSplice, based on CNN to classify candidate splice junctions.
[60]	CCDS, NCBI, TISdb datasets	CNN	One-hot Encoding	Accuracy, Sensitivity, Specificity, MCC	Proposed a DL architecture based on CNN for prediction of translation initiation sites.
[62]	Human, Cross-species, New pre-miRNAs	RNN	Word Embeddings	Sensitivity, Specificity, F-score, g-mean	Proposed a novel miRNA precursor prediction algorithm that can successfully reflect structural characteristics of precursor miRNAs.
[63]	Public human miRNA-mRNA pairing database	RNN	One-hot Encoding	Accuracy, Sensitivity, Specificity, F-score	Presents a DL framework for miRNA target prediction using deep recurrent neural networks-based autoencoding.
[64]	MAQC-III	RNN	One-hot Encoding	Pearson, Spearman, r^2	Proposed a novel approach to model nucleotide sequences to calculate sequence-specific bias without pre determining sequence structures.
[65]	GENCODE Release 25	RNN-GRU	Word Embeddings	Accuracy, Sensitivity, Specificity, F-score, PPV, NPV	Implemented a gated RNN architecture that learns complex and long range patterns for predicting protein coding potential.
[66]	Ensembl human genome and UCSC-hg38 dataset	RNN	One-hot Encoding	Accuracy	Proposed a deep RNN based approach to classify a sequence is to be a coding or noncoding sequence.
[69]	HS3D database	CNN-BLSTM	One-hot Encoding	Accuracy, Sensitivity, Specificity, MCC	Proposed a hybrid architecture for splice site classification.
[70]	GENCODE dataset	CNN-RNN	One-hot Encoding	Accuracy, loss, R^2	Proposed a computational model, COSSMO, that accurately predicts competitive alternative splice site selection from raw DNA sequences.

Summary

The review of the literature on studies related to prediction of splice site junctions indicated that DL architectures were varying between RNN and CNN. Complex hybrid architectures were also proposed that required human expertise to design them through trial and error approaches. Observations guided the choice of methods, experiments and metrics used in this thesis.

In the reviewed studies, it was observed that selection of an appropriate DL architecture according to the capabilities and characteristics of the input data is crucial for success. This required further extensive tuning of the architecture parameters to better suit the problems in hand. Many architectures were adapted from computer vision to genomics by human experts by means of parameter sharing, multi-modal, multi-task and transfer learning approaches. The success of these architectures in computer vision is based on the notion that the object to be classified occupies a considerable part of the input image. However, in genomics the classification feature occupies only a tiny fraction the input. Also DL architectures are over-parametrized and the performance can be conditional if the models are not appropriately designed.

Despite the successful adaptation of models from other domains for solving problems in genomics, transfer learning approaches still pose challenges. The best practices for training and fine tuning the models are not concrete and depend on additional hyper-parameters for specific DL architectures. The task relatedness in multi-task learning and the similarity in data and domain in transfer learning methods are difficult to assess. The differences in domains requires fine tuning or pre-training the model from scratch and it does not guarantee expected results. In genomics domain, there is a compelling need to adopt methods to discover domain-specific architectures to address specific problems.

In the context of the literature review, the following choices were made:

1. The DARTS method was adapted to discover new high performance DL architecture for DNA splice site classification. The continous approach based on DARTS approach was preferred as the NAO was proposed very recently.

2. The search space will be performed on CNN architectures as it was most used and performed well in recent studies.

3. The discovered model will be compared against fixed baseline architectures of CNN and RNN.

4. The following classification metrics are selected from the literature review to be computed: Accuracy, Sensitivity, Specificity, F score, AUC score

5. The discovered model will be investigated further to benchmark on computational performance.

CHAPTER 3: METHODOLOGY

In this chapter, we describe our representation of the input DNA sequence in Section 3.1. We then present the computational procedure for finding the best architecture cell in Section 3.2. Section 3.3 presents the methods and metrics used for the evaluation of the discovered architecture.

DNA Representation

The sequence data is biologically described using four types of nucleotides, adenine (A), cytosine (C), guanine (G) and thymine (T). Each of these sequences are converted into numerical representation using 1-dimensional orthogonal encoding or one-hot encoding for downstream analysis. However, to shape the input appropriately for the DARTS CNN model, the DNA sequences are represented as a 3-dimensional tensor, which is similar in shape to an image tensor input to image classification networks. Firstly, one-hot encoding is applied which converts each nucleotide in the DNA sequence of length n_d into a four-dimensional vector and then concatenates each of them to form the complete sequence. The next step is to transform the list of one-hot vectors to a 3-dimensional tensor.

let s ϵ S where S ={A,T,C,G}, then, a sequence (A,C,G,T,A,C) will be encoded into a tuple of 4-D binary vectors as shown in Fig. 3.1.

([1,0,0,0],[0,0,1,0],[0,0,0,1],[0,1,0,0],[1,0,0,0],[0,0,1,0])

The encoded sequence is then represented as a three-dimensional matrix of shape (3 x n_d x 4). The final representation of the input to the model will be in the form (batch size x 3 x n_d x 4)

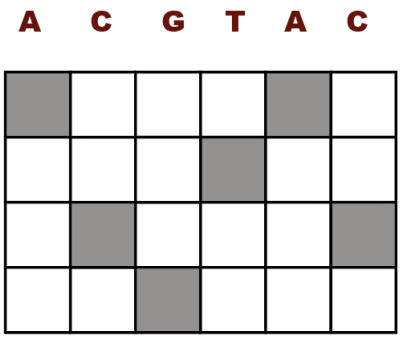


Figure 3.1: DNA representation by one-hot vector.

DARTS Algorithm

The DARTS technique discovers state-of-the art network architectures by formulating the task in a differential manner. The interesting part in this method is that the search space is treated as continuous rather than searching over a discrete set of architectures in the search space.

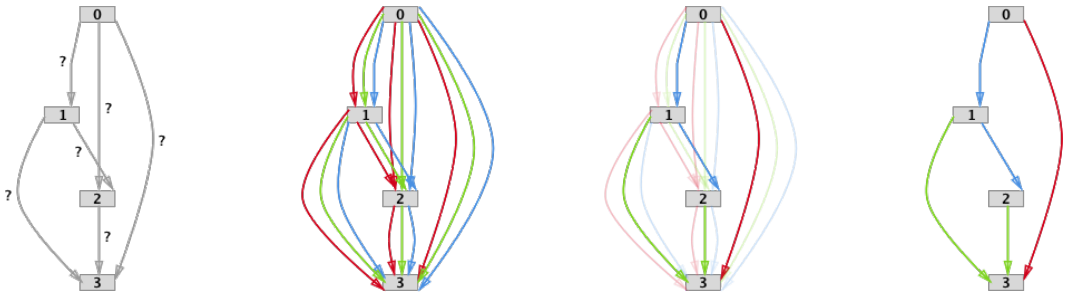


Figure 3.2: DARTS method.

The cell in the architecture is considered as a Direct Acyclic Graph (DAG) consisting of a set of nodes and edges. Each cell has one output node and two input nodes. Let N be the number of nodes and each node represented by x^i . Each edge (i, j) performs an operation represented by $o^{(i,j)}$ that transforms x^i. The intermediate nodes are computed based on its predecessors.

$$x^i = \sum_{j<i}(o^{i,j}x^j) \tag{3.1}$$

The learning of the cell involves learning the operations that transform the input. Fig. 3.2 shows an overview of the DARTS method [6]. The goal of the method is to find a cell that forms the building block of the final architecture. Initially the operations on the edges are unknown. Let O be the set of operations where each operation is represented as $o(.)$ to be applied to x^i. The choice of the operation is made in a continuous manner by performing a softmax on all possible operations.

$$o^{-(i,j)}(x) = \sum_{o \epsilon O} \frac{\exp(\alpha_0^{(i,j)})}{\sum_{o' \epsilon O} \exp(\alpha_{0'}^{()})} o(x) \tag{3.2}$$

Here $\alpha_0^{(i,j)}$ is a vector with dimension $|O|$ that indicates the mixing of operation between a pair of nodes. The architecture search phase jointly performs learning on a set of continuous variables $\alpha = \{\alpha^{(i,j)}\}$ and the weights(ω) within each operation in O. The value of α and ω is obtained through a bi-level optimization algorithm where α becomes the higher level variable and ω acts as the lower level variable. The search finds a value of α that minimizes the validation loss L_{val} for that value of ω that minimizes the training loss L_{train}.

$$\min_{\alpha} \qquad L_{valid}(\omega^*(\alpha), \alpha) \tag{3.3}$$

$$s.t \quad \omega^*(\alpha) = argmin_{\omega} \quad L_{train}(\omega, \alpha) \tag{3.4}$$

This bi-level optimization algorithm 1 shows the optimization of ω and α in the respective search spaces through a gradient-based approach. The operation at each edge is replaced by the operation that had the maximum value of α.

The optimization is performed in the architecture search algorithm 2 during the training phase and the best discrete architecture $arch_{final}$ is saved to be evaluated on the

Algorithm 1: Optimization algorithm of ω w.r.t α

Let $o^{-(i,j)}$ be the set of operations parameterized by $\alpha^{(i,j)}$ for an edge $pair(i,j)$
while *not converged* **do**
    compute ω by decreasing $\nabla_{\omega} L_{train}(\omega, \alpha)$;
    compute α by decreasing $\nabla_{\alpha} L_{valid}(\omega - \xi \nabla_{\omega} L_{train}(\omega, \alpha), \alpha)$;
Replace $o^{-(i,j)}$ with $o^{(i,j)}$= $argmax_{o \epsilon O}\ \alpha_o^{i,j}$ for each pair of edge(i,j)

architecture evaluation algorithm 3. The training phase in the architecture evaluation phase has a fixed architecture $arch_{final}$ and is then trained to obtain the optimal architecture weights. The trained model is evaluated on unseen data in the architecture evaluation phase.

Algorithm 2: Architecture Search Algorithm

Phase1: Architecture Search
Input : x_s=($x_{s,1}$,$x_{s,2}$,$x_{s,3}$,...,$x_{s,|x_s|}$) $x_s \epsilon X_s$
where X: a set of sequences with $|X|$=N
x_s: a single sequence with length $|x_s|$
$x_{(s,i)} \epsilon \{A, T, C, G\}$ for $i = 1, 2, 3....|x_s|$
y: label for x_s. $y \epsilon \{0, 1\}$
where 0 means non-splice and 1 represents splice sequence
Output: $arch_{final}$
where $arch_{final}$ is the final architecture cell
`/* Data Preprocessing and Loading */`
$X_T \leftarrow transform(X_s)$; where X_T: Transformed dataset (Section 3.1)
$(x_t, x_v) \leftarrow split(X_T)$; where x_t: Training set and x_v: Validation set
initialize best accuracy acc_b ;
for *each epoch* **do**
    `/* Training */`
    for m,n *training data randomly selected from* x_t,x_v **do**
        initialize new architecture $arch(\omega, \alpha)$ using Algorithm 1;
        train($arch(\omega, \alpha)$); ; `// Training the model`
    `/* Validation */`
    for p *validation data randomly selected from* x_v **do**
        $acc_{val} = valid(arch(\omega, \alpha))$; ; `// Validating the model`
    if $acc_{val} > acc_b$ **then**
        acc_b=acc_{val};
        $arch_{final} = arch(\omega, \alpha)$;
save $arch_{final}$ for architecture evaluation phase;

Algorithm 3: Architecture Evaluation Algorithm

Phase2: Architecture Evaluation

Input : x_s=($x_{s,1}$,$x_{s,2}$,$x_{s,3}$,...,$x_{s,|x_s|}$) $x_s \epsilon X_s$
where X: a set of sequences with $|X|$=N
x_s: a single sequence with length $|x_s|$
$x_{(s,i)} \epsilon \{A, T, C, G\}$ for $i = 1, 2, 3....|x_s|$
y: label for x_s. $y \epsilon \{0, 1\}$
where 0 means non-splice and 1 represents splice sequence

Output: acc_{final}
where acc_{final} is the final accuracy

`/* Data Preprocessing and Loading */`

$X_T \leftarrow transform(X_s)$; where X_T: Transformed dataset (Section 3.1)

$(x_t, x_v, x_f) \leftarrow split(X_T)$; where x_t: Training set, x_v: Validation set and x_f: Test set

initialize best accuracy acc_b ;

initialize weights ω;

initialize architecture $arch$ to $arch_{final}$ obtained from Algorithm 2;

for *each epoch* **do**

`/* Training */`

for m *training data randomly selected from* x_t **do**

train($arch(\omega, \alpha)$); ; `// Training the model`

update weights ω using SGD with momentum ;

`/* Validation */`

for p *validation data randomly selected from* x_v **do**

$acc_{val} = valid(arch(\omega, \alpha))$; ; `// Validating the trained model`

if $acc_{val} > acc_b$ **then**

acc_b=acc_{val};

save best model $arch(\omega, \alpha)$;

acc_{final}=test($arch(\omega, \alpha)$); ; `// Testing the final model`

Performance Evaluation Method

The performance of DARTS model was compared using the metrics described in the Section 3.31 and Section 3.3.2. The results proved that DARTS model outperformed other fixed DL architectures. The benchmarking metrics were used to evaluate the model on multiple GPU and CPU architectures.

Classification Performance Metrics

There are several metrics used for assessing the classification performance. These metrics have been derived from the confusion matrix as shown in Fig. 3.3. The terminologies used in the confusion matrix are described in the following

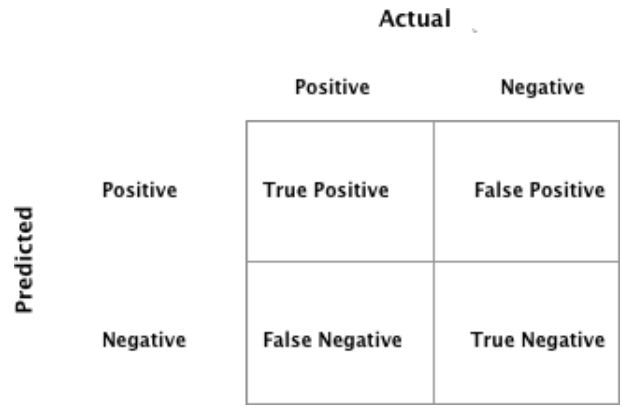


Figure 3.3: Confusion matrix.

- **Condition Positives (P)**: The total identified number of samples that have the feature.
- **Condition Negatives (N)**: The total identified number of samples that does not have the feature.
- **True Positives (TP)**: The total identified number of samples predicted to have the feature, and in fact they have the feature.
- **True Negatives (TN)**: The total identified number of samples predicted to not have the feature, and in fact they do not have the feature.
- **False Positives (FP)**: The total identified number of samples predicted to have the feature, and in fact they do not have the feature.
- **False Negatives (FN)**: The total identified number of samples predicted to not have the feature, and in fact they have the feature.

- **Accuracy**: The percentage of samples that were correctly classified by the model

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.5}$$

- **Sensitivity**: The sensitivity is the measure of the model to correctly classify samples with the feature.

$$Sensitivity = \frac{TP}{TP + FN} \tag{3.6}$$

- **Specificity**: The specificity is the measure of the model to correctly classify samples without the feature.

$$Specificity = \frac{TN}{TN + FP} \tag{3.7}$$

- **F-score**: The weighted average of the measure of precision and recall.

- **Area Under the ROC Curve(AUC)**: The measure of the performance of the model across all possible classification thresholds.

Computational Performance Metrics

The resources in high-performance computing (HPC) are formed by a heterogeneous hardware. The GPU is a parallel computing coprocessor that are specialized in accelerating vector computations. Training of DL models are mostly performed on GPU due to large number of vector computations that take place and CPU is used for scalar operations. However, the CPU-based benchmarking is included for users that run DL workloads in a CPU environment. The performance of DARTS model was benchmarked on multiple GPU and CPU architectures and the time taken for architecture search and architecture evaluation on each were noted. The learning

and inference speed of the trained model was compared on the different GPU architectures over three types of datatype: single precision, double precision and half precision.

CHAPTER 4: IMPLEMENTATION

The methodology makes use of deep CNN to distinguish features between true and false splice junctions. CNN architectures have shown better performance in learning features that classify actual splice sites from false ones. The method consists of two stages: Architecture Search and Architecture Evaluation.In the first stage, architecture search using DARTS was performed to discover the best model and the second stage validates the discovered model on a held out unseen data. The model gains from the information present in the genomic sequence of the candidate splice junction to accurately classify whether the sequence corresponds to a splice junction or not.

Dataset

The experiments were performed using Splice Site Recognition (SSR) dataset that is publicly available at [71]. The underlying problem posed in this dataset is to classify, given a sequence of DNA, as a splice or a non-splice sequence. The splice junctions are locations in a DNA sequence where 'superfluous' DNA is removed during protein creation process. The beginning and end of an exon is determined by the splice-donor and splice-acceptor sequences present. In this study, the prediction of splice junctions are performed using the given annotated DNA sequences with true acceptor splice site sequences. The original dataset has 159,771 true acceptor splice site sequences and 14,868,555 non-acceptor sequences. To avoid the class imbalance problem we used all the true acceptor splice site sequences and equal number of non-acceptor sequences chosen randomly. The dataset used in our study has a total of 319542 DNA sequence samples with a sequence length of 141 base pair. The experiments were repeated 10 times for better performance validation and robustness of the trained model. In each of the experiment, the dataset was randomly split

into three sets from the complete dataset using customized data loaders into train, validation and test datasets respectively of size 0.6, 0.2, 0.2.

Experimental Setup

The architecture search and evaluation experiments were performed on Sidra HPC environment using NVIDIA Tesla K40M. The implementation was done using Pytorch which is an open-source ML library in python based on Torch that supports strong GPU acceleration.

The computational benchmarking experiments were performed on Intel(R) Xeon(R) x86_64, Quadro K4100M and Tesla V100-SXM2 architectures. The specification of the hardware architectures used is listed in Table 4.1. The GPUs and CPUs were configured with the same environment as the original experiments. The CPU evaluation was performed by submitting the task as a job to the cluster environment.

Table 4.1: Hardware for Benchmarking

Device Model	**Number of available cores**	**Memory size(in GB)**
Tesla V100-SXM2	5120	16
Tesla K40m	2880	12
Quadro K4100M	1152	4
Intel(R) Xeon(R) CPU E5-2670	32	256

Architecture Search

The following operations were only included from a rich primitive space used in [6] for our search space O :3x3 separable convolutions, 3x3 dilated separable convolutions, 3x3 max pooling, 3x3 average pooling and zero. A building block of a convo-

lution operation is formed by three steps: Firstly an activation function is Rectifier Linear Unit (RELU) applied and then a convolution operation (CN) is executed and finally, a batch normalization (BN) is performed. This is denoted as ReLU-Conv-BN and we used the same ReLU-Conv-BN order in [6] for performing convolutional operations. Our discovered convolutional cell consisted of 4 nodes, where the output node is result of the depth-wise concatenation of the convolution and pooling layers excluding the input node. The final architecture network was formed by stacking multiple cells together. The architecture consists of two types of convolutional cells called normal cell and reduction cell to make it scalable for any input size. When a feature map is taken as input, the normal cell returns a feature map of same dimension. The reduction cell returns a feature map where the height and width are reduced by a factor of two. The reduction cells are located at 1/3 and 2/3 of the total depth of the architecture. The architecture has a reduction cell in every third cell of the complete architecture. The first and second input nodes of the cell k are set to the $k-2$ and $k-1$ cells respectively. The Figures 4.1 to 4.4 shows the discovered normal and reduction cells during the first 4 epochs in the architecture search experiment.

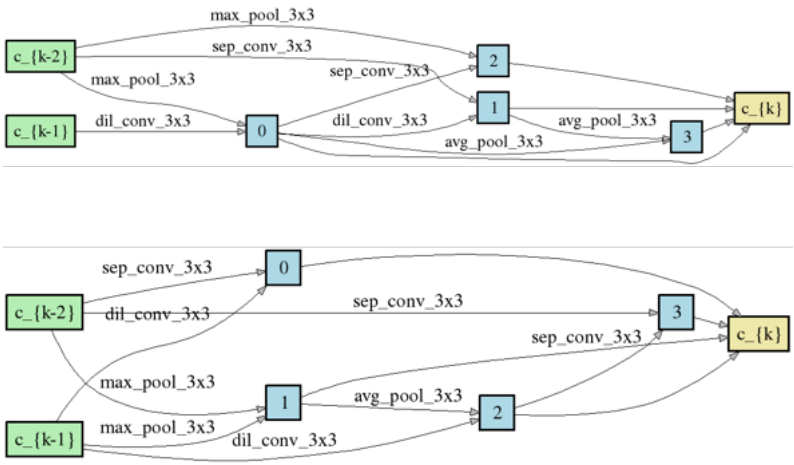


Figure 4.1: Normal and Reduction cell in Epoch 1.

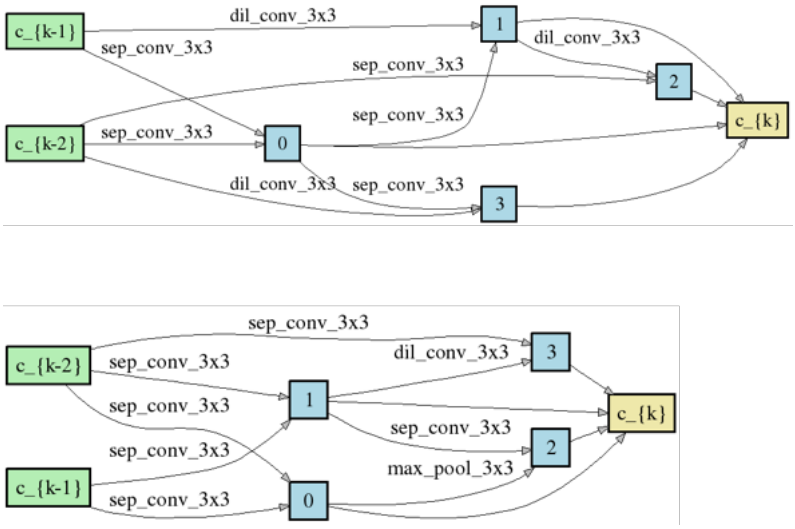


Figure 4.2: Normal and Reduction cell in Epoch 2.

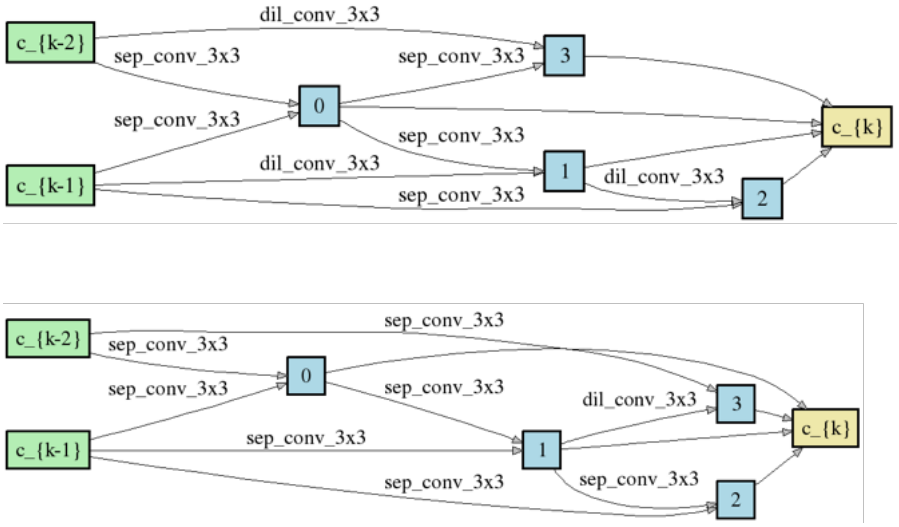


Figure 4.3: Normal and Reduction cell in Epoch 3.

The architecture search was performed on train and validation datasets. A network composed of 8 cells were trained for 50 epochs using DARTS with batch size 500 set for both training and validation. The weights ω were optimized using SGD with momentum and Adam as the optimizer for architecture variables. The initial learning rate was set as 0.0025 and was gradually decreased to a minimum of 0.001. The rest

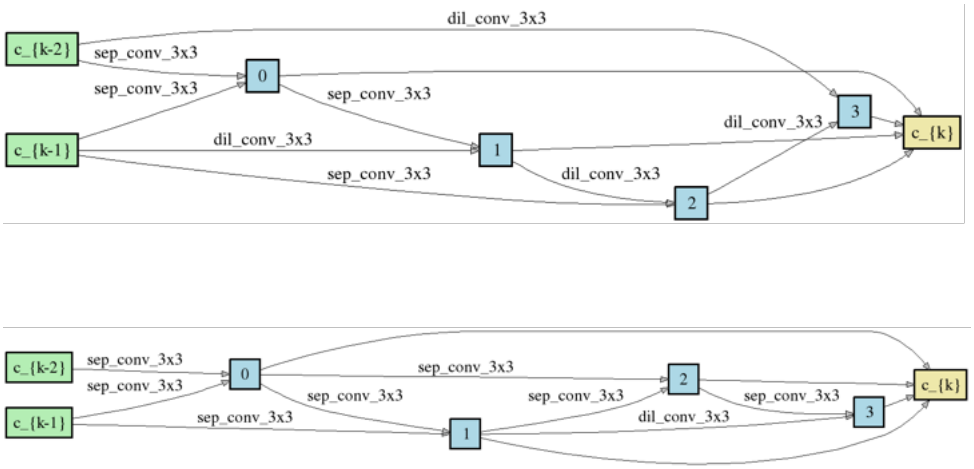


Figure 4.4: Normal and Reduction cell in Epoch 4.

of the hyperparameters were chosen similar to the original implementation in [6] as shown in Table 4.2.

Table 4.2: Hyperparameters for architecture search

Hyperparameter	Definition	Value
Batch size	Number of samples propagated through the network	500
Initial learning rate	The initial rate which influences the current value of weights of the model during training	0.0003
Architecture learning rate	The learning rate for architecture encoding	0.0025
Minimum learning rate	The minimal value to which the learning rate will decrease during model training	0.001
Epochs	Number of training iterations of the model	50
Weight decay rate	A regularization rate to improve the generalization performance of neural networks	0.0003
Loss function	The function used to evaluate the set of weights	Cross Entropy
Update strategy	The optimization technique used to update the model parameters	SGD with momentum

Architecture Evaluation

The best architecture cells shown in Fig. 4.5 were selected based on the performance on validation dataset. In order to select the optimal architecture for evaluation, the search experiment was run ten times with different seeds. The best performing cell was recorded during epoch 29 in the ninth iteration. A network of 3 cells were trained

for 40 epochs with batch size 500. The rest of the hyperparameters were similar to the ones used for the architecture search process. The selected best architecture was evaluated using a held out test dataset. It is important to note that the test set was never used during the architecture search process. As the results were subject to variance even for same configurations, the mean and standard deviation of 10 independent runs was reported.

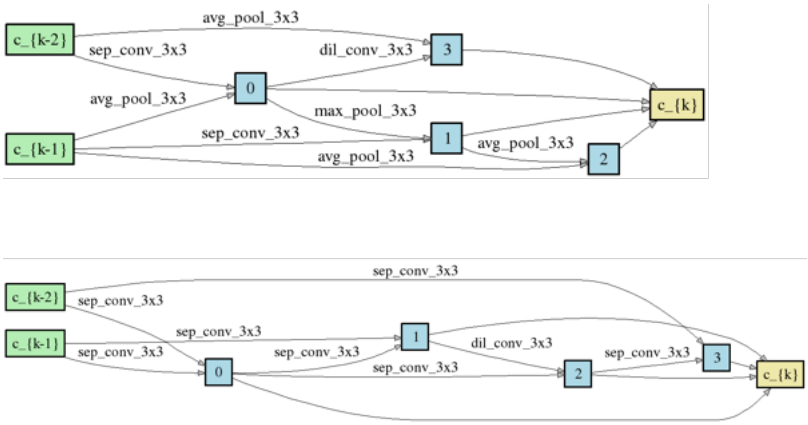


Figure 4.5: Normal and Reduction cell learned on splice dataset.

CHAPTER 5: RESULTS

In this chapter we present the results of the benchmarking experiments conducted. Section 5.1 presents the classification performance of DARTS model with five established models in DL. In Section 5.2, the computational performance of the model is benchmarked on GPU and CPU.

Classification Performance Results

The splice dataset classification results using various DL techniques are presented in Table 5.1. The average and standard deviation metrics were calculated over 10 repetitions of each experiment. The DARTS experiment was repeated by running the experiments on 10 different seeds. The experiments on the baseline models were repeated by k-fold cross-validation. The performance of the automatically discovered architecture was compared against fixed baseline CNN and RNN models. The baseline models were implemented using keras framework to benchmark against known models from the literature. Notably, DARTS outperformed over the widely known LSTM and baseline CNN with and without embedding in terms of test accuracy, sensitivity, specificity, F-score and AUC score. The model achieved better accuracy results over the hybrid and LSTM model with embedding.

Table 5.1: Comparison of Model Performance

Model	Accuracy	Sensitivity	Specificity	F-score	AUC score
DARTS	**94.15 ± 0.12**	94.00 ± 0.10	95.20± 0.30	94 ± 0.10	94.8 ± 0.01
Baseline LSTM with embeddings	93.98 ± 0.11	95 ± 1.09	92.32 ± 1.25	93.74 ± 0.25	93.66 ± 0.27
Hybrid(RNN+CNN)	93.66 ± 0.27	94.63 ± 0.78	93.33 ± 0.63	94.02 ± 0.14	93.66 ± 0.27
Baseline LSTM	86.99 ± 14.65	85.26 ± 19.81	88.71 ± 9.80	85.87 ± 16.93	86.99 ± 14.65
Baseline CNN	64.15 ± 0.63	65.27 ± 6.00	63.04 ± 6.22	64.43 ± 2.16	64.10 ± 0.50
Baseline CNN with embeddings	53.63 ± 3.65	95.62 ± 6.61	11.63 ± 13.77	67.35 ± 0.51	53.63 ± 3.65

Fig. 5.1 and Fig. 5.2 shows the mean and standard deviation plots of training and validation metrics over 40 epochs for all the baseline models against our proposed model. Notably, the fixed CNN baseline architectures performed poorly compared to other architectures.

Computational Performance Results

Execution Time

In addition, the proposed model was benchmarked on GPU systems and CPU. The execution time for performing architecture search and architecture evaluation were calculated on different device architectures and the results are presented in Table 5.2. Notably, the most advanced Tesla V100 GPU completed the search in less than 11 hours and the evaluation in half an hour. Fig. 5.3 shows the plot of search and evaluation execution time of DARTS model on each of the devices.

Table 5.2: Comparison of Execution Time

Device Model	**Architecture Search(hours)**	**Architecture Evaluation(hours)**
Tesla V100-SXM2	10.75	0.5
Tesla K40m	38.5	2
Quadro K4100M	101	50
Intel(R) Xeon(R) CPU E5-2670	526	74

Precision

The learning and inference speed of the trained model were compared on different GPU architectures. The experiments were performed on single precision, half precision, double precision data types. The model was fed with a single batch input of 500 sequences. For training, the time required for 20 forward and backward passes

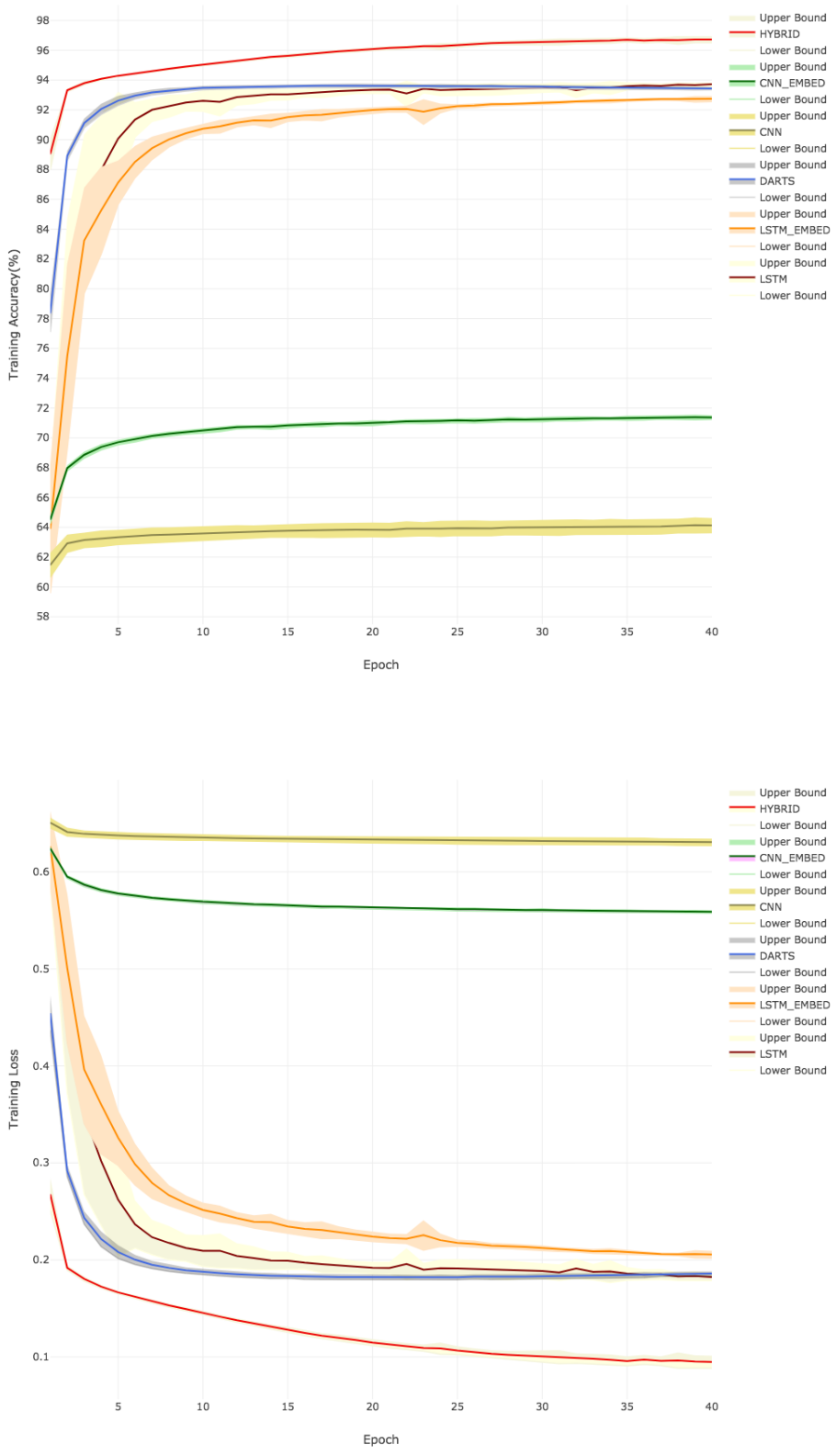


Figure 5.1: Plots of training accuracy and loss.

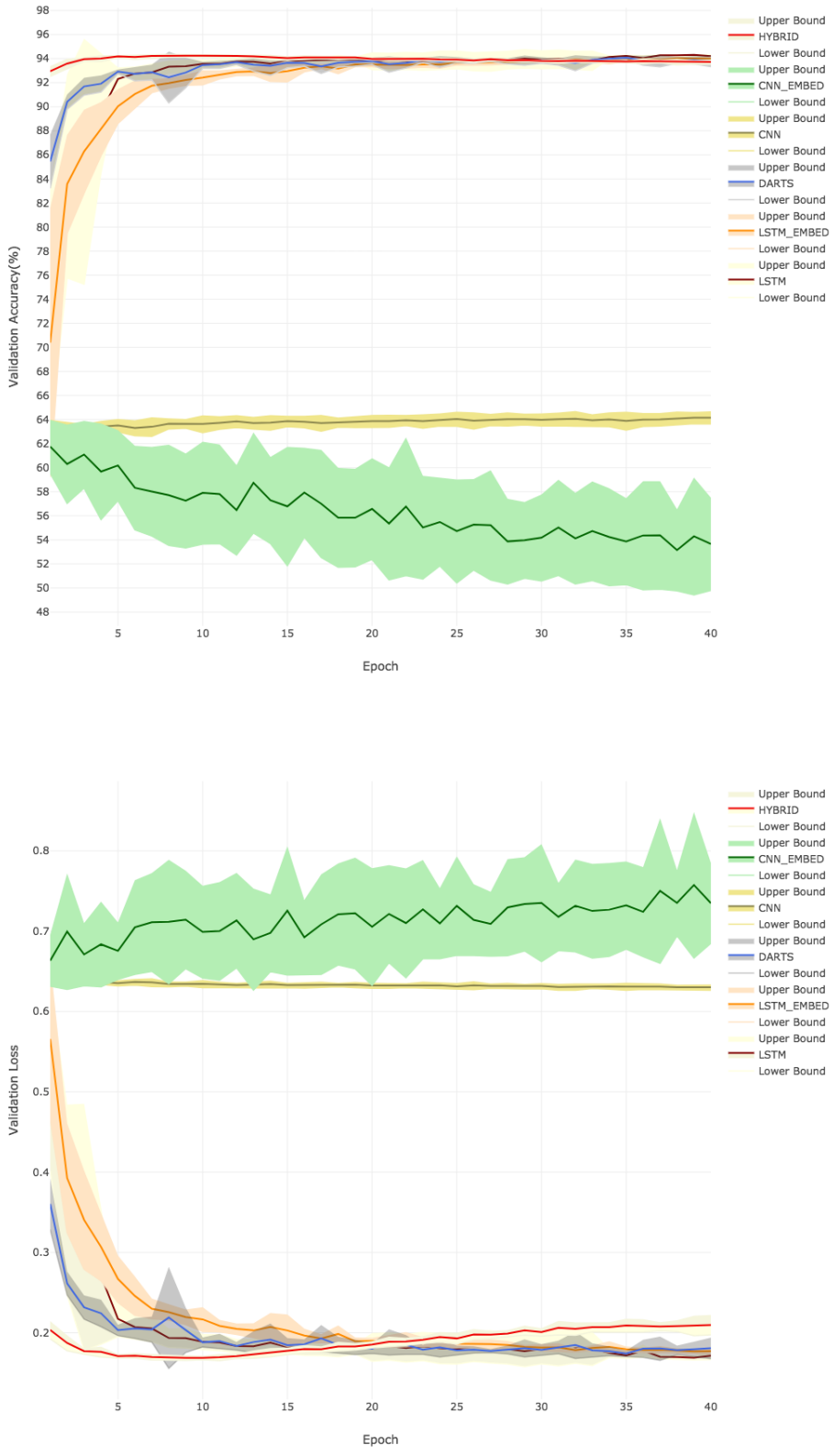


Figure 5.2: Plots of validation accuracy and loss.

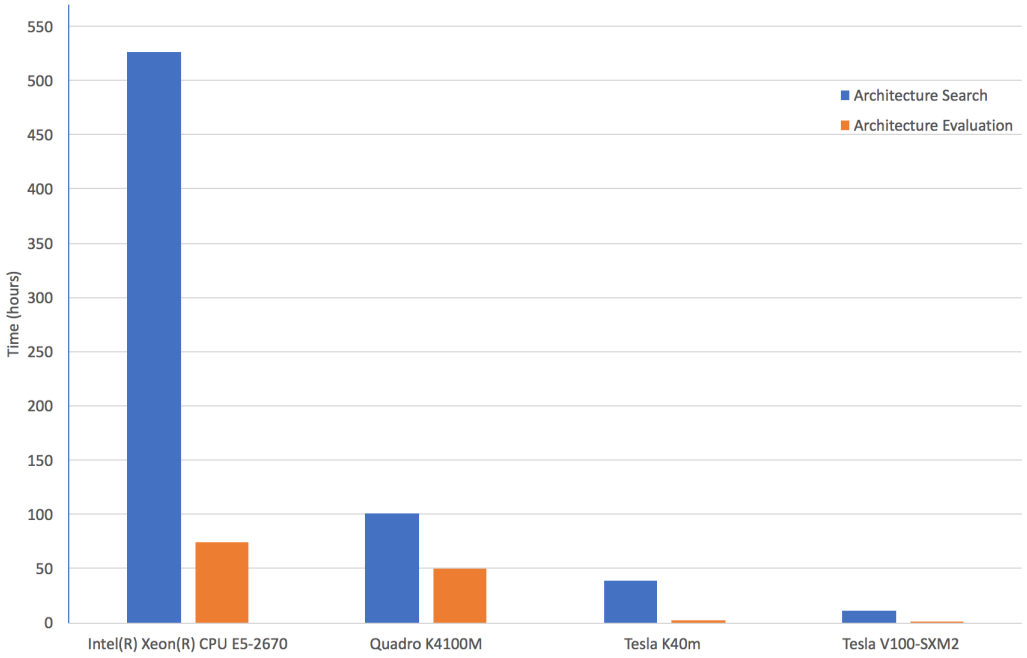


Figure 5.3: Plot of execution time for DARTS.

were averaged. In inference, time duration of 20 forward passes were averaged. Five warm up steps were included that was not calculated towards the final results. Fig. 5.4 shows the plot of training and inference speed of DARTS model on each of the devices.

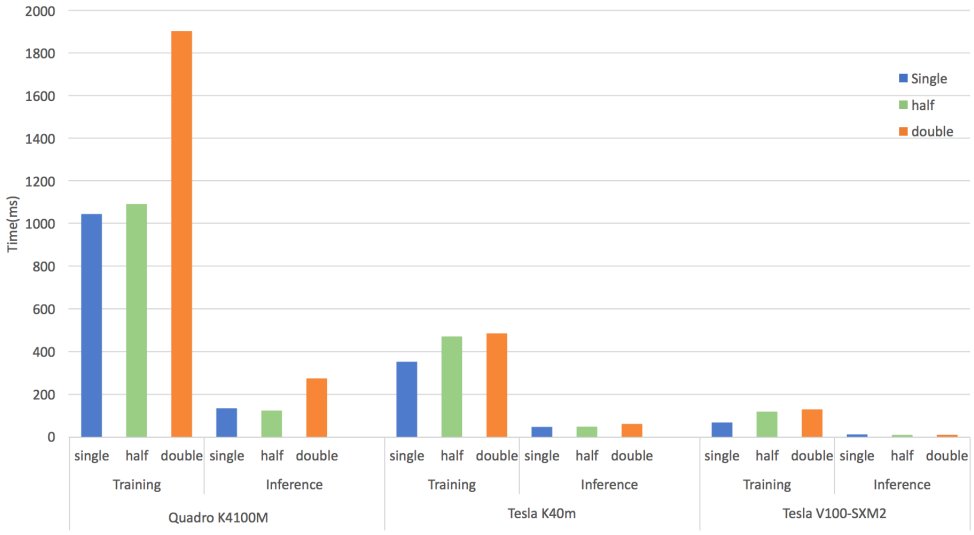


Figure 5.4: Plot of comparison of training and inference speed.

Table 5.3: Comparison of Learning and Inference Speed on GPUs

GPU Model	Training			Inference		
	Single	Half	Double	Single	Half	Double
Quadro K4100M	1041.24	1090.38	1902.38	134.97	124.59	274.78
Tesla K40m	352.53	470.72	485.17	48.022	49.38	62.13
Tesla V100-SXM2	68.46	119.55	130.09	13.21	11.02	11.07

CHAPTER 6: CONCLUSION AND FUTURE WORK

Deep Learning is an emerging research topic among the genomics community. Its applications can be revolutionized by introducing high-performance computing methods to analyze datasets in the field of gene therapies, molecular diagnostics and personalized medicine. In the scope of this thesis, an advanced DL approach based on differential architecture search was implemented to solve the splice site classification problem in genomics and to discover new high performance CNN architectures. It was observed in the literature review that most of the work was focused on using manually designed architectures in ML and DL. This study has aided in bridging the gap between the state-of-the art in DL and its application to genomics. The evaluation results showed that the newly discovered architecture outperformed the fixed baseline DL architectures using the same dataset. The architecture was compared alongside the well-known LSTM model and complex hybrid architectures. Furthermore, the discovered architecture was evaluated on multiple CPU and GPU architectures. The total time taken for performing the architecture search and evaluation were determined as well as the floating point instructions per second for single, double and half precision were compared. The computational benchmarking results obtained proved that there is significant improvement in execution time when using advanced GPU architectures.

For all its promises, DL in genomics still possess a number of challenges. The results largely depend on the quality of the data input that are well annotated so that the model can learn to distinguish features and identify patterns. Another challenge is the lack of judgement capability where the technique is able to distinguish from a biologically relevant variation and normal variations. This would require applying further experimental design and controls. The advancements in the field of DL in the field of computer vision and speech recognition has led to new methods being constantly proposed that awaits its application in genomics domain. Furthermore, the

availability of quasi-unlimited storage at a reasonable price, the surge in computing power and the lower computational costs will allow these advanced DL techniques to reshape the capabilities of machines to completely understand and interpret the human genome.

As future steps, the plan is to further improve the performance of DARTS based on CNN by including more primitives such as skip connect and higher convolutional operations, thereby widening the architecture search space. This will help to traverse more information to lower layers. The DARTS approach will further be evaluated against the recent parallel work of NAO which also performs continuous optimization of architecture space. The study showed that fixed RNN architectures have better results than CNN. Genomics data are sequential data similar to speech recognition, natural language processing and language translation. It would be interesting to implement DARTS to search for a recurrent cell that can be recursively connected to form a RNN that can be applied for tasks of protein function prediction. As part of performance benchmarking, the plan is to evaluate the models further on GPU clusters and Google's Tensor Processing Unit (TPU) which are specialized hardware architectures developed to accelerate AI workloads. In addition, this approach will be tested on future genomics classification tasks, as it will be highly useful to uncover new insights from the vast available sequencing data.

REFERENCES

[1] J. D. Watson, F. H. Crick, *et al.*, "Molecular structure of nucleic acids," *Nature*, vol. 171, no. 4356, pp. 737–738, 1953.

[2] P. V. Jithesh and V. Scaria, "From genomes to genomic medicine: enabling personalized and precision medicine in the middle east," *Personalized Medicine*, vol. 14, no. 5, pp. 377–382, 2017. PMID: 29754564.

[3] 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," *Journal of The Royal Society Interface*, vol. 15, no. 141, p. 20170387, 2018.

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

[5] 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*, pp. 770–778, 2016.

[6] H. Liu, K. Simonyan, and Y. Yang, "Darts: Differentiable architecture search," *arXiv preprint arXiv:1806.09055*, 2018.

[7] P. Baldi, P. Sadowski, and D. Whiteson, "Searching for exotic particles in high-energy physics with deep learning," *Nature communications*, vol. 5, p. 4308, 2014.

[8] G. B. Goh, N. O. Hodas, and A. Vishnu, "Deep learning for computational chemistry," *Journal of computational chemistry*, vol. 38, no. 16, pp. 1291–1307, 2017.

[9] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, p. 115, 2017.

[10] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[11] Y. Bengio *et al.*, "Learning deep architectures for ai," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[12] M. K. Leung, H. Y. Xiong, L. J. Lee, and B. J. Frey, "Deep learning of the tissue-regulated splicing code," *Bioinformatics*, vol. 30, no. 12, pp. i121–i129, 2014.

[13] H. Y. Xiong, B. Alipanahi, L. J. Lee, H. Bretschneider, D. Merico, R. K. Yuen, Y. Hua, S. Gueroussov, H. S. Najafabadi, T. R. Hughes, *et al.*, "The human splicing code reveals new insights into the genetic determinants of disease," *Science*, vol. 347, no. 6218, p. 1254806, 2015.

[14] B. E. Bejnordi, M. Veta, P. J. Van Diest, B. Van Ginneken, N. Karssemeijer, G. Litjens, J. A. Van Der Laak, M. Hermsen, Q. F. Manson, M. Balkenhol, *et al.*, "Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer," *Jama*, vol. 318, no. 22, pp. 2199–2210, 2017.

[15] J. Lanchantin, R. Singh, Z. Lin, and Y. Qi, "Deep motif: Visualizing genomic sequence classifications," *arXiv preprint arXiv:1605.01133*, 2016.

[16] J. Zhou and O. G. Troyanskaya, "Deep supervised and convolutional generative stochastic network for protein secondary structure prediction," *arXiv preprint arXiv:1403.1347*, 2014.

[17] D. R. Kelley, J. Snoek, and J. L. Rinn, "Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks," *Genome research*, vol. 26, no. 7, pp. 990–999, 2016.

[18] H. Zeng, M. D. Edwards, G. Liu, and D. K. Gifford, "Convolutional neural network architectures for predicting dna–protein binding," *Bioinformatics*, vol. 32, no. 12, pp. i121–i127, 2016.

[19] B. Alipanahi, A. Delong, M. T. Weirauch, and B. J. Frey, "Predicting the sequence specificities of dna-and rna-binding proteins by deep learning," *Nature biotechnology*, vol. 33, no. 8, p. 831, 2015.

[20] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

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

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

[23] V. Boža, B. Brejová, and T. Vinař, "Deepnano: deep recurrent neural networks for base calling in minion nanopore reads," *PloS one*, vol. 12, no. 6, p. e0178751, 2017.

[24] D. Quang and X. Xie, "Danq: a hybrid convolutional and recurrent deep neural network for quantifying the function of dna sequences," *Nucleic acids research*, vol. 44, no. 11, pp. e107–e107, 2016.

[25] A. Busia, J. Collins, and N. Jaitly, "Protein secondary structure prediction using deep multi-scale convolutional neural networks and next-step conditioning," *arXiv preprint arXiv:1611.01503*, 2016.

[26] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of machine learning research*, vol. 11, no. Dec, pp. 3371–3408, 2010.

[27] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," in *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pp. 833–840, Omnipress, 2011.

[28] C. Poultney, S. Chopra, Y. L. Cun, *et al.*, "Efficient learning of sparse representations with an energy-based model," in *Advances in neural information processing systems*, pp. 1137–1144, 2007.

[29] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[30] L. Rampasek, D. Hidru, P. Smirnov, B. Haibe-Kains, and A. Goldenberg, "Dr. vae: Drug response variational autoencoder," *arXiv preprint arXiv:1706.08203*, 2017.

[31] T. Chen, R. Xu, Y. He, and X. Wang, "A gloss composition and context clustering based distributed word sense representation model," *Entropy*, vol. 17, no. 9, pp. 6007–6024, 2015.

[32] S. Hijazi, R. Kumar, and C. Rowen, "Using convolutional neural networks for image recognition," *Cadence Design Systems Inc.: San Jose, CA, USA*, 2015.

[33] 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*, pp. 1–9, 2015.

[34] H. Liu, K. Simonyan, O. Vinyals, C. Fernando, and K. Kavukcuoglu, "Hierarchical representations for efficient architecture search," *arXiv preprint arXiv:1711.00436*, 2017.

[35] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," *arXiv preprint arXiv:1802.01548*, 2018.

[36] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," *arXiv preprint arXiv:1611.01578*, 2016.

[37] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," *arXiv preprint arXiv:1707.07012*, vol. 2, no. 6, 2017.

[38] P. D. Summers, "A methodology for lisp program construction from examples," *Journal of the ACM (JACM)*, vol. 24, no. 1, pp. 161–175, 1977.

[39] B. Baker, O. Gupta, R. Raskar, and N. Naik, "Accelerating neural architecture search using performance prediction," *arXiv preprint arXiv:1705.10823*, 2017.

[40] A. Brock, T. Lim, J. M. Ritchie, and N. Weston, "Smash: one-shot model architecture search through hypernetworks," *arXiv preprint arXiv:1708.05344*, 2017.

[41] H. Pham, M. Y. Guan, B. Zoph, Q. V. Le, and J. Dean, "Efficient neural architecture search via parameter sharing," *arXiv preprint arXiv:1802.03268*, 2018.

[42] R. Luo, F. Tian, T. Qin, E. Chen, and T.-Y. Liu, "Neural architecture optimization," in *Advances in neural information processing systems*, pp. 7816–7827, 2018.

[43] B. Lee, T. Lee, B. Na, and S. Yoon, "Dna-level splice junction prediction using deep recurrent neural networks," *arXiv preprint arXiv:1512.05135*, 2015.

[44] K. F. Au, H. Jiang, L. Lin, Y. Xing, and W. H. Wong, "Detection of splice junctions from paired-end rna-seq data by splicemap," *Nucleic acids research*, vol. 38, no. 14, pp. 4570–4578, 2010.

[45] C. Trapnell, L. Pachter, and S. L. Salzberg, "Tophat: discovering splice junctions with rna-seq," *Bioinformatics*, vol. 25, no. 9, pp. 1105–1111, 2009.

[46] A. K. Baten, B. C. Chang, S. K. Halgamuge, and J. Li, "Splice site identification using probabilistic parameters and svm classification," in *BMC bioinformatics*, vol. 7, p. S15, BioMed Central, 2006.

[47] P. K. Meher, T. K. Sahu, A. Rao, and S. Wahi, "Identification of donor splice sites using support vector machine: a computational approach based on positional, compositional and dependency features," *Algorithms for molecular biology*, vol. 11, no. 1, p. 16, 2016.

[48] Y. Zhang, C.-H. Chu, Y. Chen, H. Zha, and X. Ji, "Splice site prediction using support vector machines with a bayes kernel," *Expert Systems with Applications*, vol. 30, no. 1, pp. 73–81, 2006.

[49] D. Wei, W. Zhuang, Q. Jiang, and Y. Wei, "A new classification method for human gene splice site prediction," in *International Conference on Health Information Science*, pp. 121–130, Springer, 2012.

[50] P. K. Meher, T. K. Sahu, and A. R. Rao, "Prediction of donor splice sites using random forest with a new sequence encoding approach," *BioData mining*, vol. 9, no. 1, p. 4, 2016.

[51] H. S. Lopes, C. R. Erig Lima, and N. J. Murata, "A configware approach for high-speed parallel analysis of genomic data," *Journal of Circuits, Systems, and Computers*, vol. 16, no. 04, pp. 527–540, 2007.

[52] U. Kamath, K. De Jong, and A. Shehu, "Effective automated feature construction and selection for classification of biological sequences," *PloS one*, vol. 9, no. 7, p. e99982, 2014.

[53] Q. Zhang, Q. Peng, Q. Zhang, Y. Yan, K. Li, and J. Li, "Splice sites prediction of human genome using length-variable markov model and feature selection," *Expert Systems with Applications*, vol. 37, no. 4, pp. 2771–2782, 2010.

[54] E. Pashaei, A. Yilmaz, M. Ozen, and N. Aydin, "Prediction of splice site using adaboost with a new sequence encoding approach," in *Systems, Man, and Cybernetics (SMC), 2016 IEEE International Conference on*, pp. 003853–003858, IEEE, 2016.

[55] R. Elsousy, N. Kathiresan, and S. Boughorbel, "On the depth of deep learning models for splice site identification," *bioRxiv*, p. 380667, 2018.

[56] X. Du, Y. Yao, Y. Diao, H. Zhu, Y. Zhang, and S. Li, "Deepss: Exploring splice site motif through convolutional neural network directly from dna sequence," *IEEE Access*, vol. 6, pp. 32958–32978, 2018.

[57] Y. Zhang, X. Liu, J. N. MacLeod, and J. Liu, "Deepsplice: Deep classification of novel splice junctions revealed by rna-seq," in *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 330–333, IEEE, 2016.

[58] N. G. Nguyen, V. A. Tran, D. L. Ngo, D. Phan, F. R. Lumbanraja, M. R. Faisal, B. Abapihi, M. Kubo, and K. Satou, "Dna sequence classification by convolutional neural network," *Journal of Biomedical Science and Engineering*, vol. 9, no. 05, p. 280, 2016.

[59] Y. Zhang, X. Liu, J. MacLeod, and J. Liu, "Discerning novel splice junctions derived from rna-seq alignment: a deep learning approach," *BMC Genomics*, vol. 19, no. 1, p. 971, 2018.

[60] J. Zuallaert, M. Kim, Y. Saeys, and W. De Neve, "Interpretable convolutional neural networks for effective translation initiation site prediction," in *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 1233–1237, IEEE, 2017.

[61] S. T. Kothen-Hill, A. Zviran, R. C. Schulman, S. Deochand, F. Gaiti, D. Maloney, K. Y. Huang, W. Liao, N. Robine, N. D. Omans, *et al.*, "Deep learning mutation prediction enables early stage lung cancer detection in liquid biopsy," 2018.

[62] S. Park, S. Min, H. Choi, and S. Yoon, "deepmirgene: deep neural network based precursor microrna prediction," *arXiv preprint arXiv:1605.00017*, 2016.

[63] B. Lee, J. Baek, S. Park, and S. Yoon, "deeptarget: end-to-end learning framework for microrna target prediction using deep recurrent neural networks," in *Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pp. 434–442, ACM, 2016.

[64] Y.-z. Zhang, R. Yamaguchi, S. Imoto, and S. Miyano, "Sequence-specific bias correction for rna-seq data using recurrent neural networks," *BMC genomics*, vol. 18, no. 1, p. 1044, 2017.

[65] S. T. Hill, R. Kuintzle, A. Teegarden, E. Merrill III, P. Danaee, and D. A. Hendrix, "A deep recurrent neural network discovers complex biological rules to decipher rna protein-coding potential," *Nucleic acids research*, vol. 46, no. 16, pp. 8105–8113, 2018.

[66] H. Bae, B. Lee, S. Kwon, and S. Yoon, "Dna steganalysis using deep recurrent neural networks," *arXiv preprint arXiv:1704.08443*, 2017.

[67] T. Lee and S. Yoon, "Boosted categorical restricted boltzmann machine for computational prediction of splice junctions," in *International Conference on Machine Learning*, pp. 2483–2492, 2015.

[68] Z.-C. Xu, P. Wang, W.-R. Qiu, and X. Xiao, "iss-pc: Identifying splicing sites via physical-chemical properties using deep sparse auto-encoder," *Scientific Reports*, vol. 7, no. 1, p. 8222, 2017.

[69] T. Naito, "Human splice-site prediction with deep neural networks," *Journal of Computational Biology*, vol. 25, no. 8, pp. 954–961, 2018.

[70] H. Bretschneider, S. Gandhi, A. G. Deshwar, K. Zuberi, and B. J. Frey, "Cossmo: predicting competitive alternative splice site selection using deep learning," *Bioinformatics*, vol. 34, no. 13, pp. i429–i437, 2018.

[71] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf, "Large scale multiple kernel learning," *Journal of Machine Learning Research*, vol. 7, no. Jul, pp. 1531–1565, 2006.

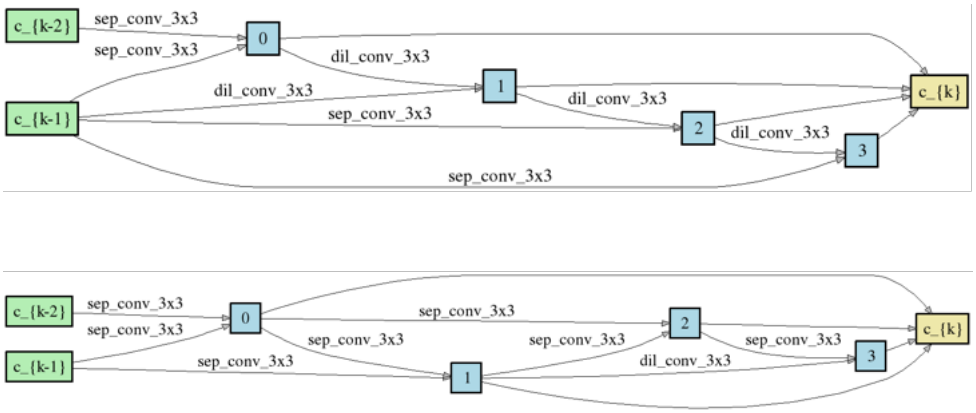


Figure A.1: Normal and Reduction cell in Epoch 6.

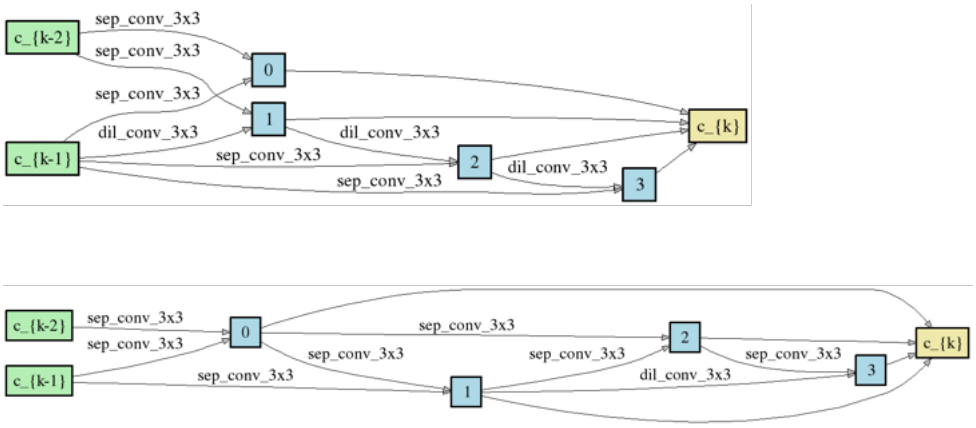


Figure A.2: Normal and Reduction cell in Epoch 7.

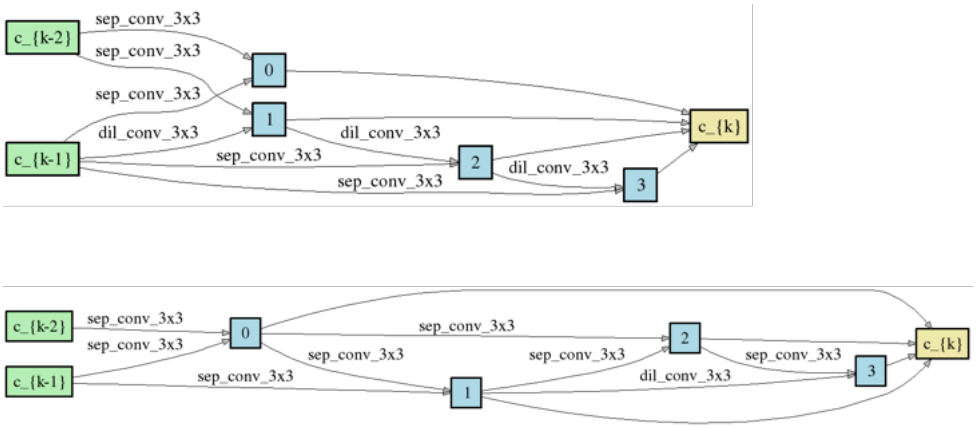


Figure A.3: Normal and Reduction cell in Epoch 8.

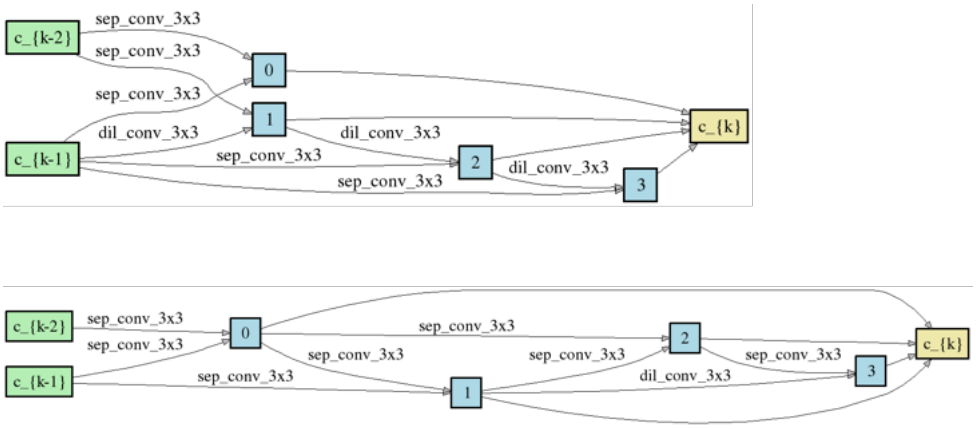


Figure A.4: Normal and Reduction cell in Epoch 9.

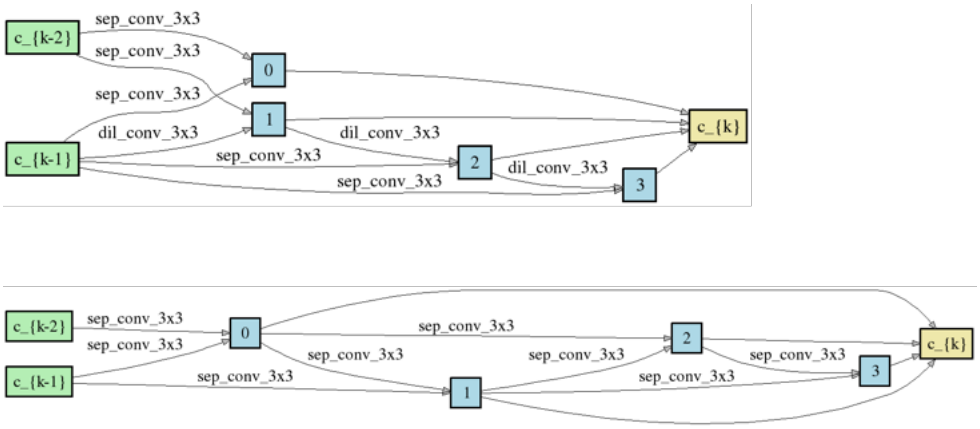


Figure A.5: Normal and Reduction cell in Epoch 10.

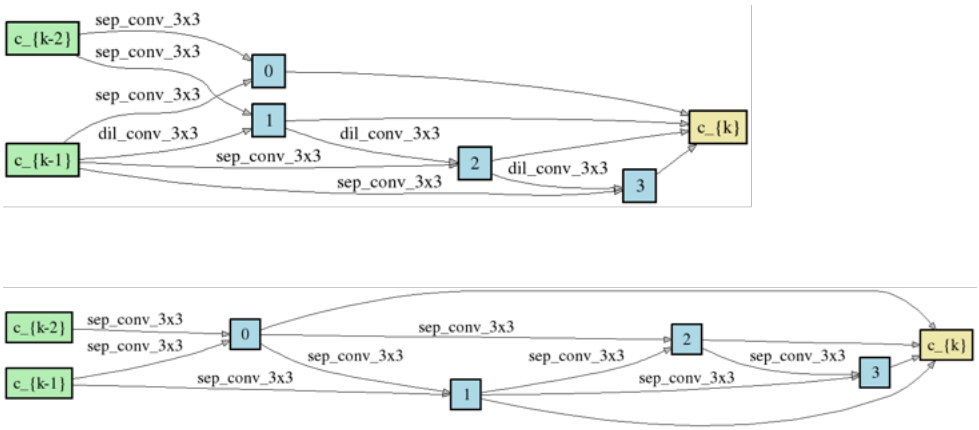


Figure A.6: Normal and Reduction cell in Epoch 11.

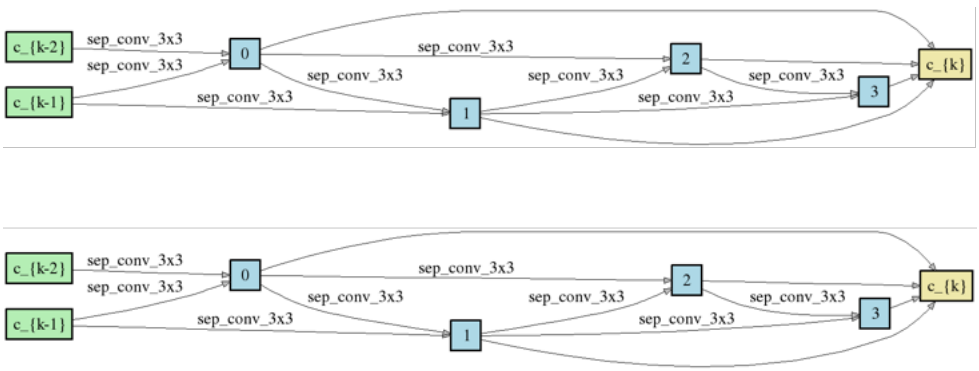


Figure A.7: Normal and Reduction cell in Epoch 12.

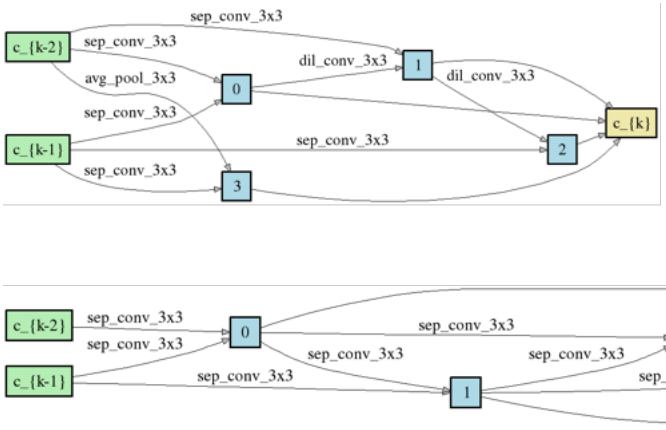


Figure A.8: Normal and Reduction cell in Epoch 13.

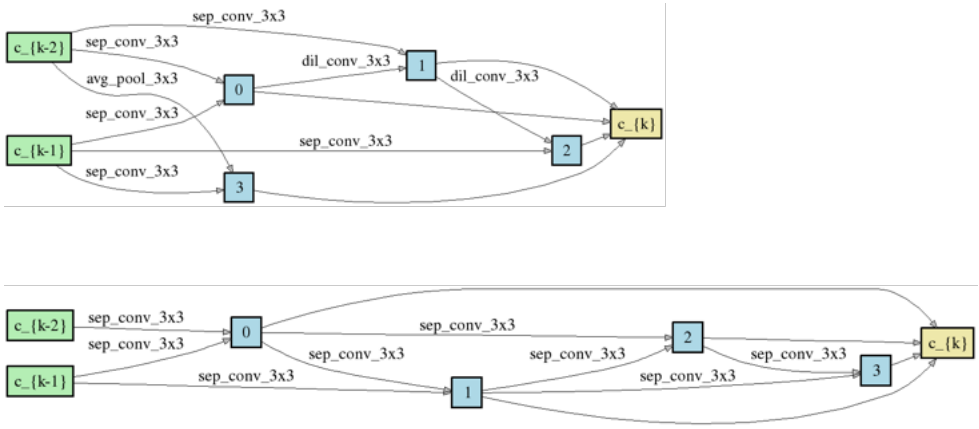


Figure A.9: Normal and Reduction cell in Epoch 14.

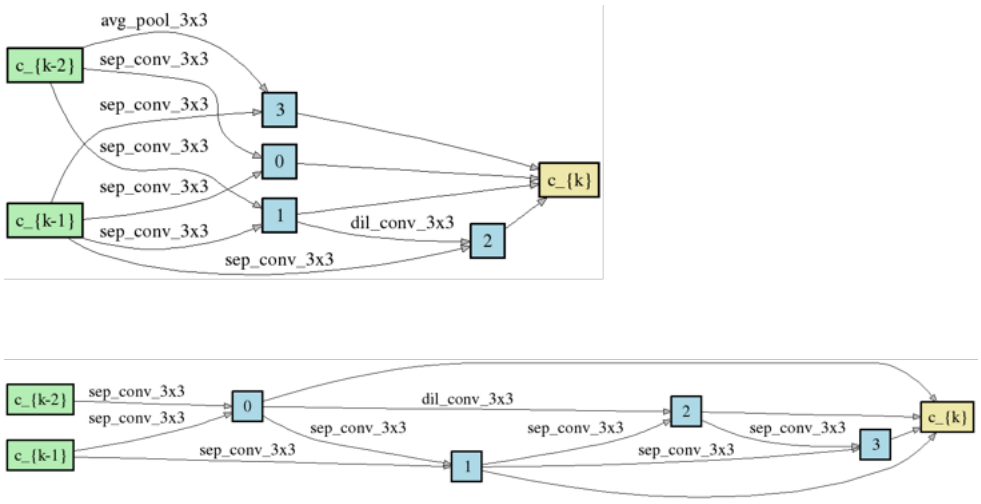


Figure A.10: Normal and Reduction cell in Epoch 15.

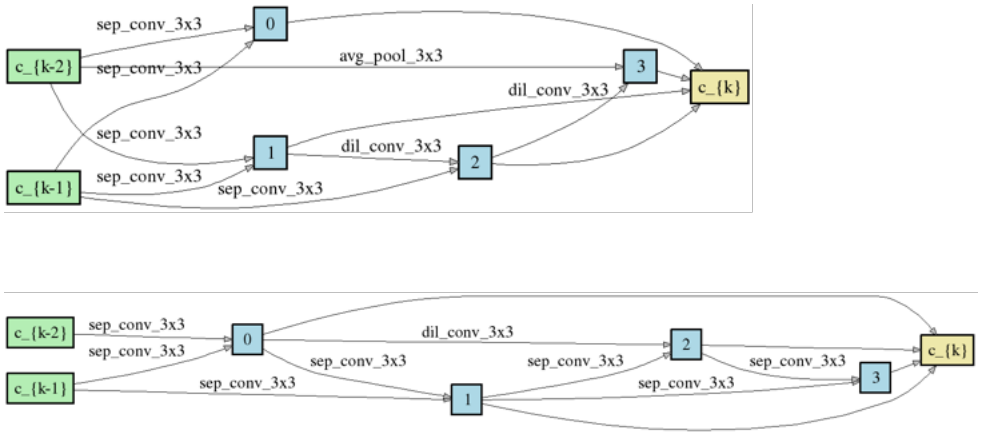


Figure A.11: Normal and Reduction cell in Epoch 16.

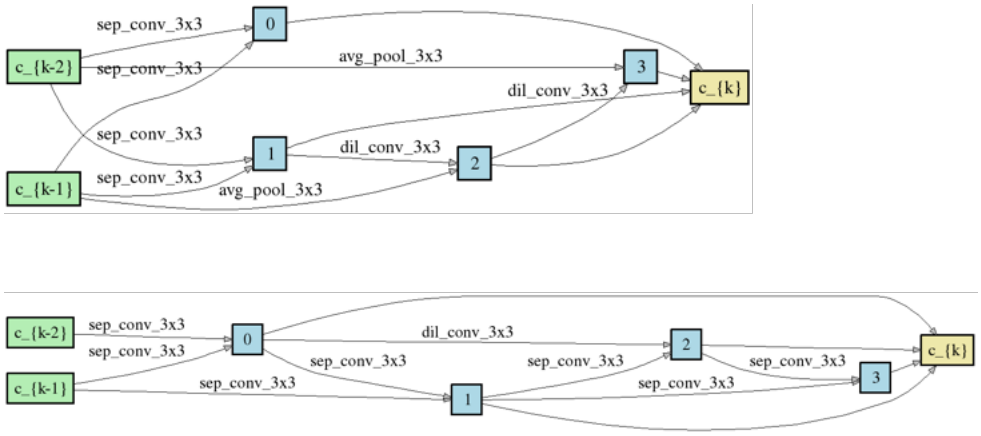


Figure A.12: Normal and Reduction cell in Epoch 17.

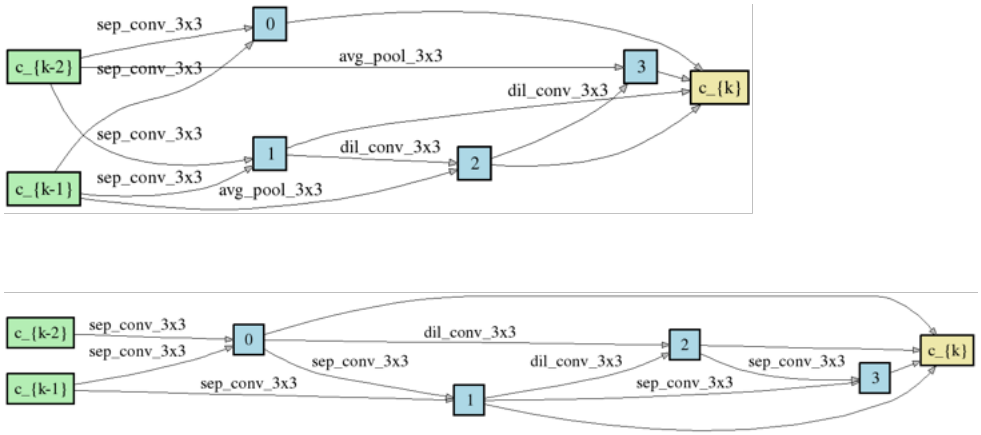


Figure A.13: Normal and Reduction cell in Epoch 18.

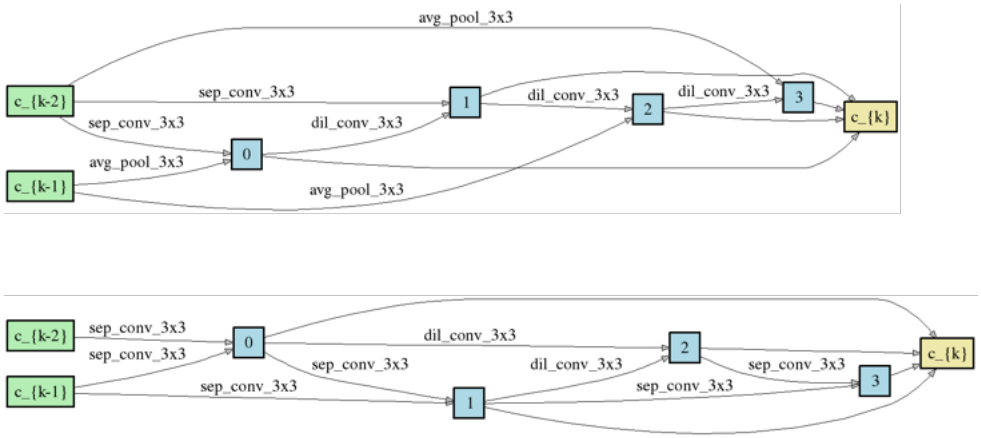


Figure A.14: Normal and Reduction cell in Epoch 19.

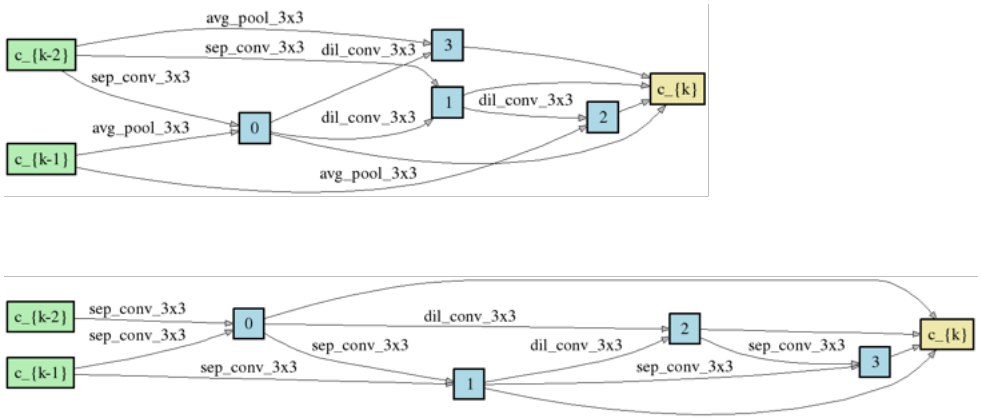


Figure A.15: Normal and Reduction cell in Epoch 20.

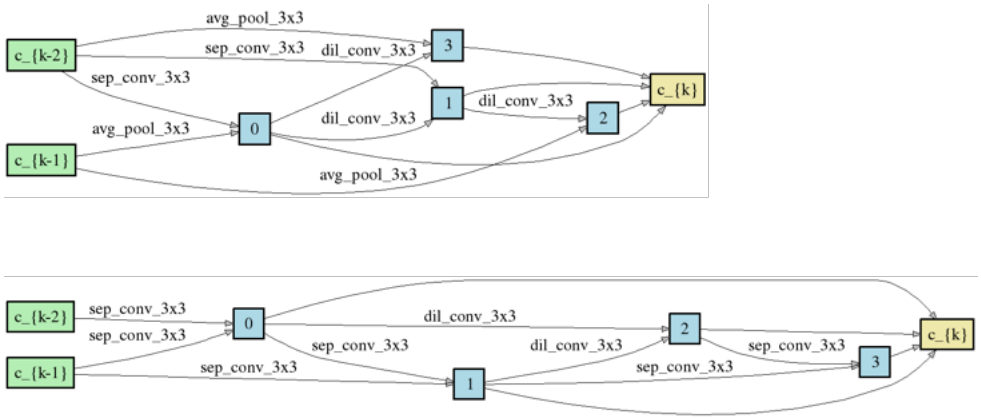


Figure A.16: Normal and Reduction cell in Epoch 21.

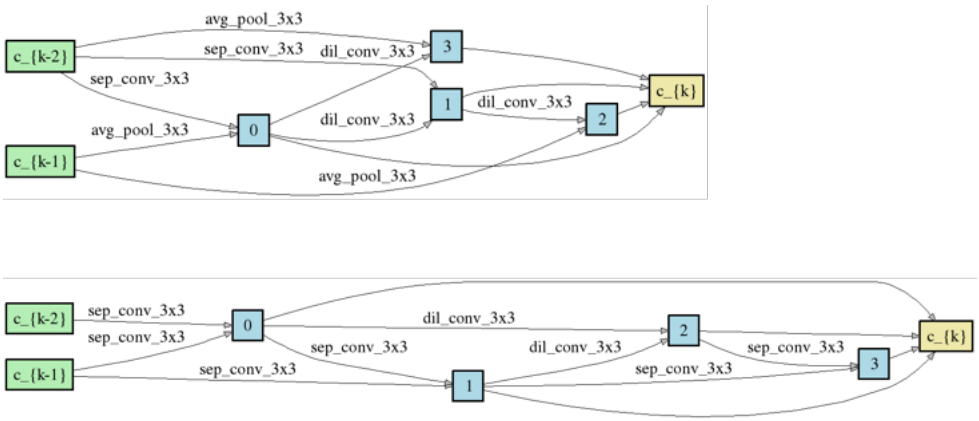


Figure A.17: Normal and Reduction cell in Epoch 22.

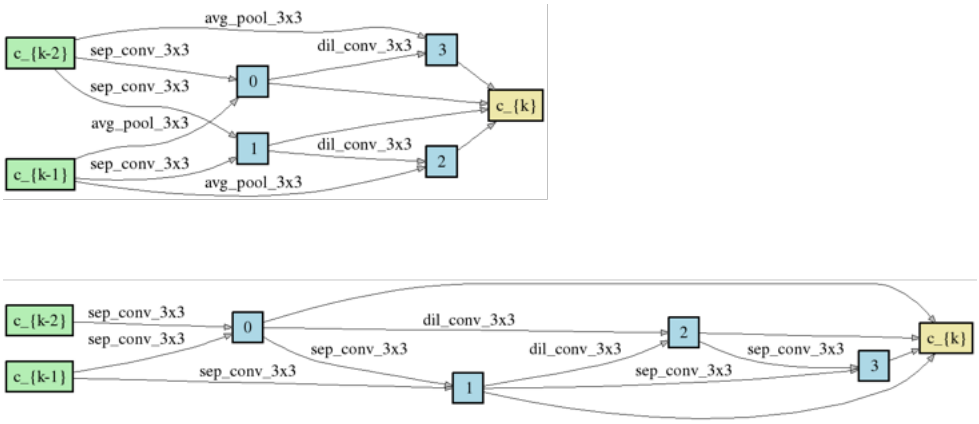


Figure A.18: Normal and Reduction cell in Epoch 23.

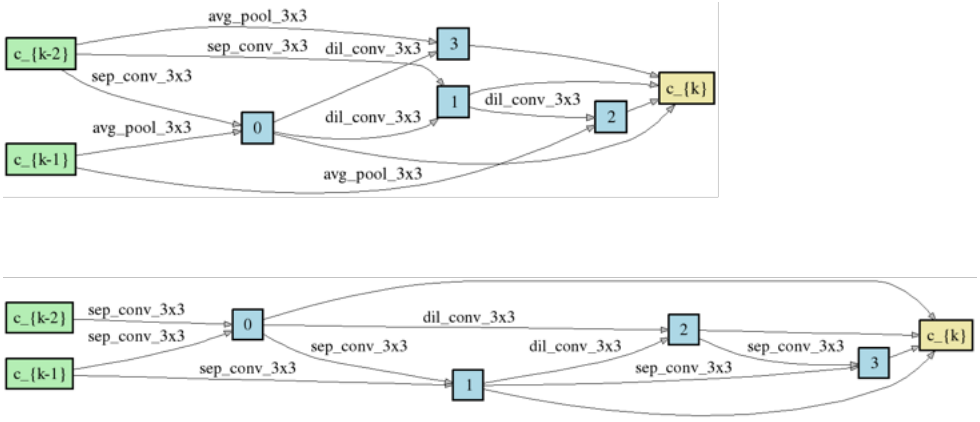


Figure A.19: Normal and Reduction cell in Epoch 24.

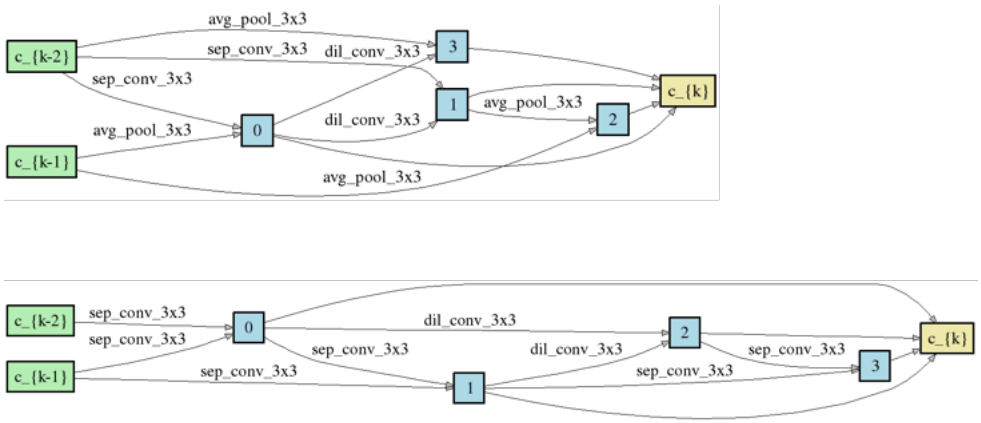


Figure A.20: Normal and Reduction cell in Epoch 25.

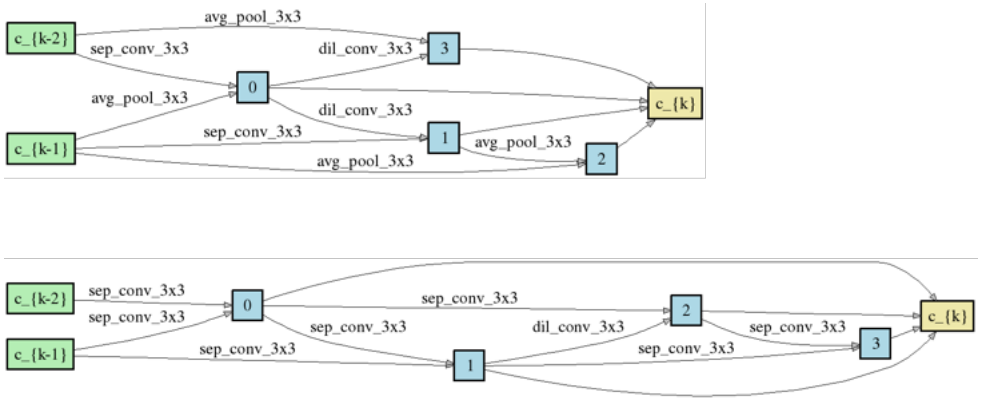


Figure A.21: Normal and Reduction cell in Epoch 26.

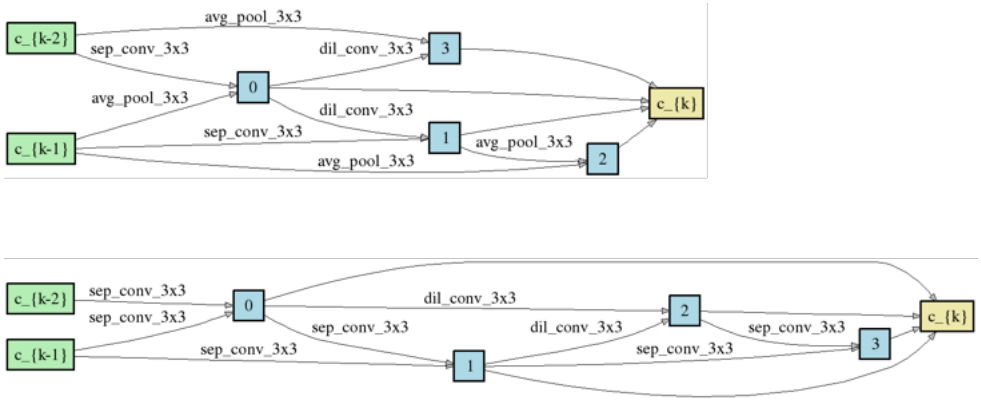


Figure A.22: Normal and Reduction cell in Epoch 27.

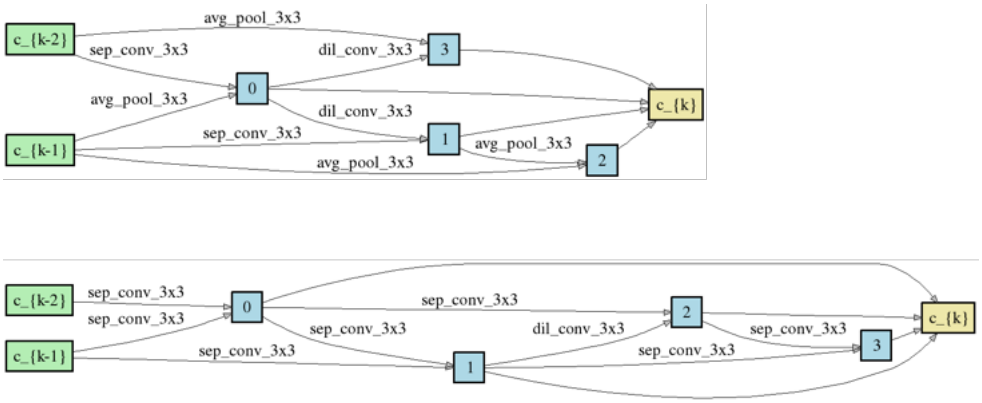


Figure A.23: Normal and Reduction cell in Epoch 28.

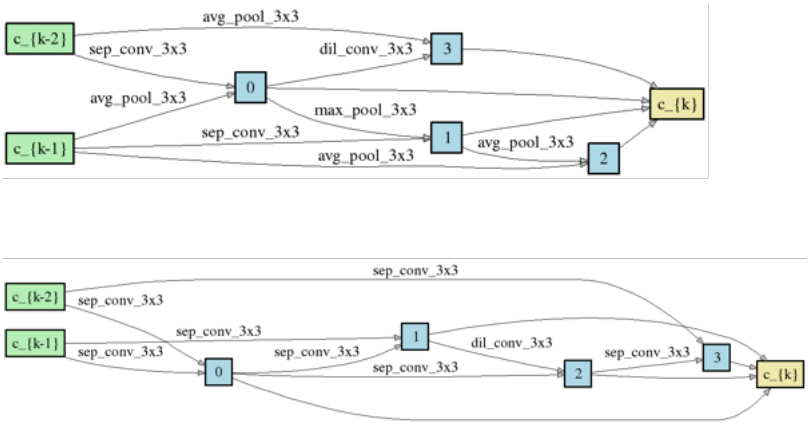


Figure A.24: Normal and Reduction cell in Epoch 29.

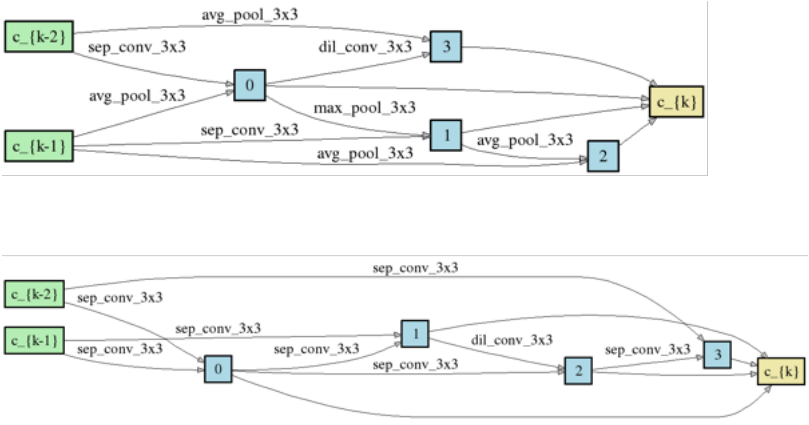


Figure A.25: Normal and Reduction cell in Epoch 30.

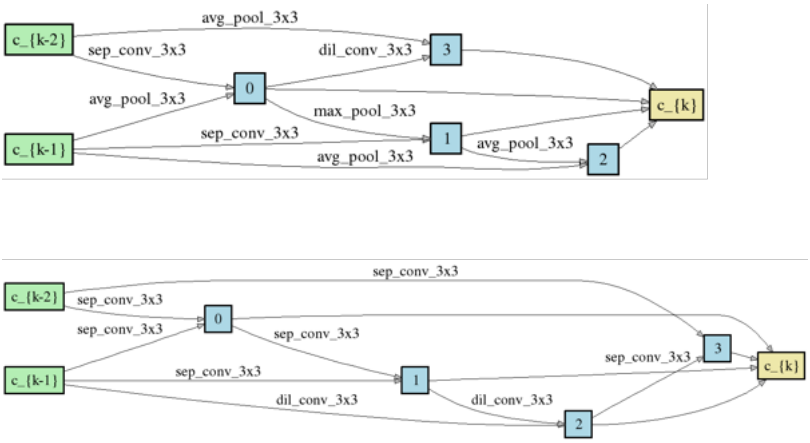


Figure A.26: Normal and Reduction cell in Epoch 31.

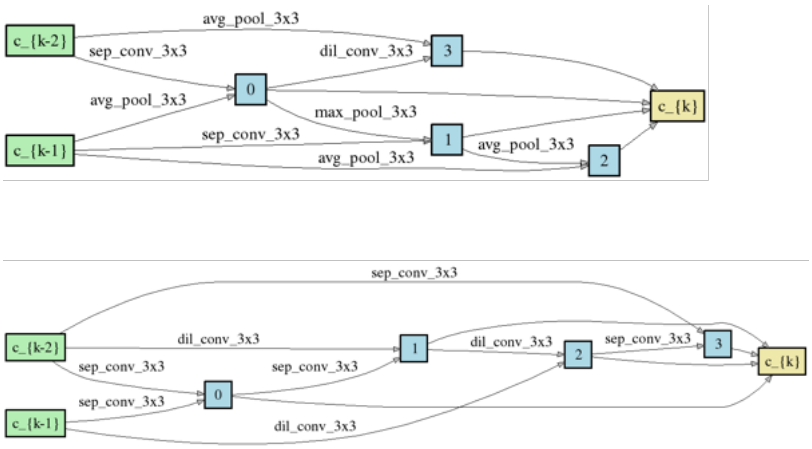


Figure A.27: Normal and Reduction cell in Epoch 32.

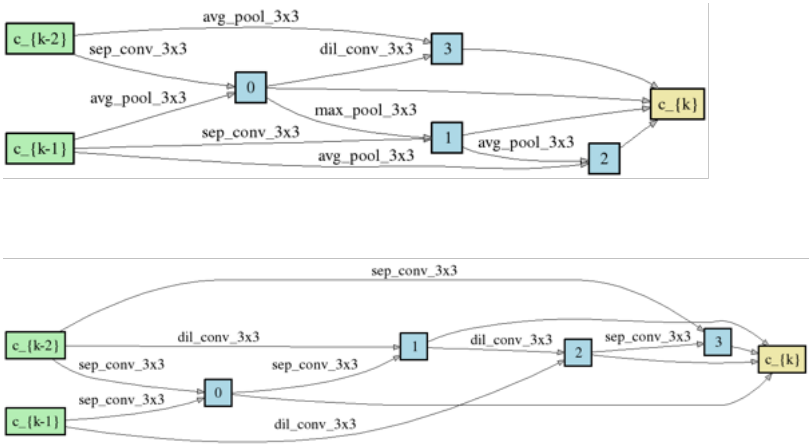


Figure A.28: Normal and Reduction cell in Epoch 33.

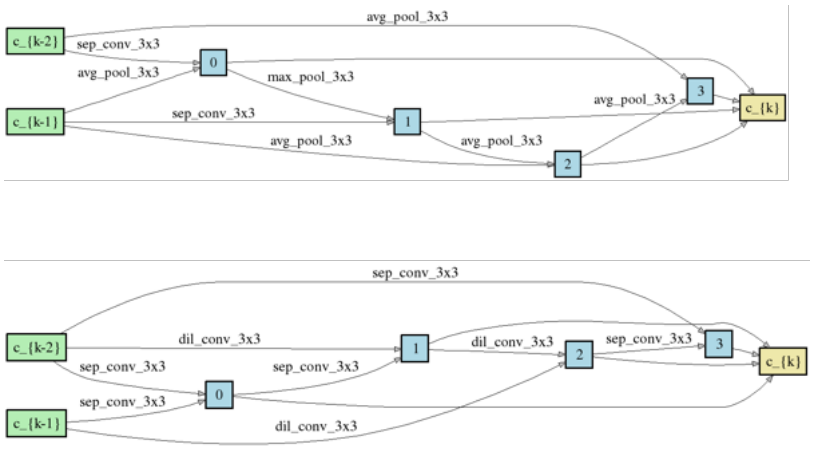


Figure A.29: Normal and Reduction cell in Epoch 34.

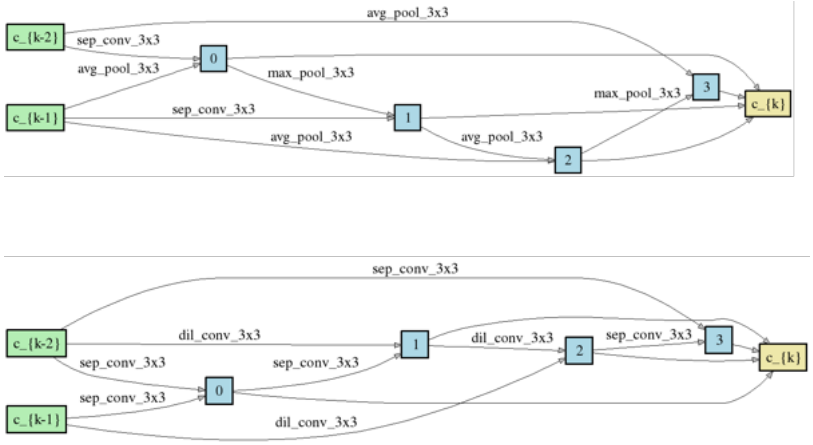


Figure A.30: Normal and Reduction cell in Epoch 35.

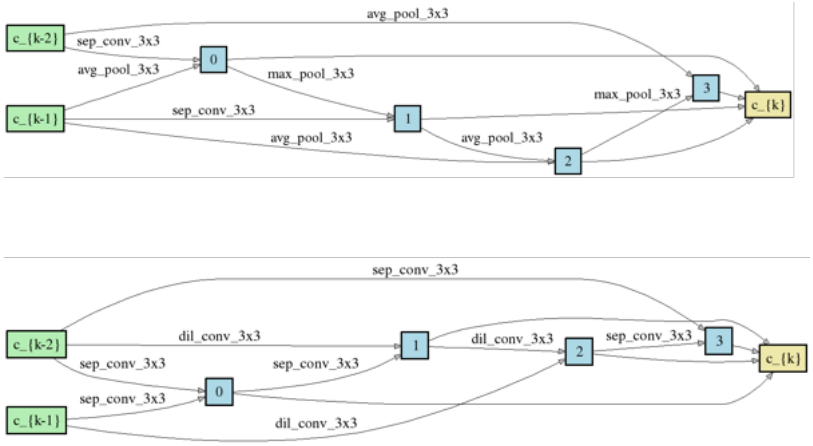


Figure A.31: Normal and Reduction cell in Epoch 36.

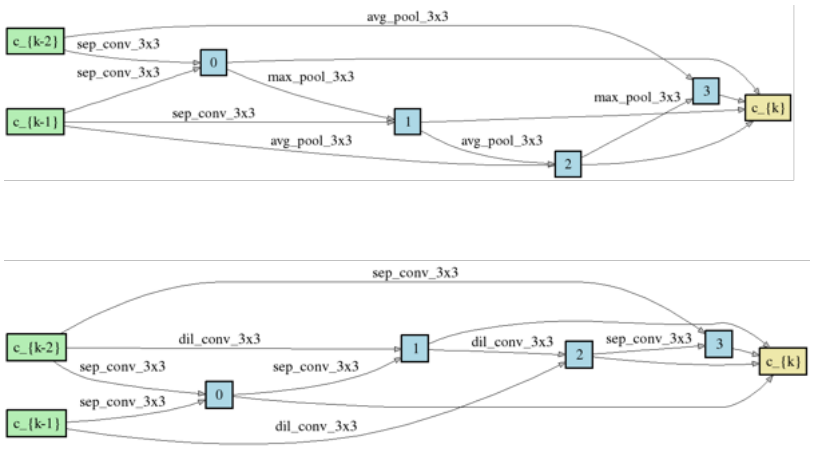


Figure A.32: Normal and Reduction cell in Epoch 37.

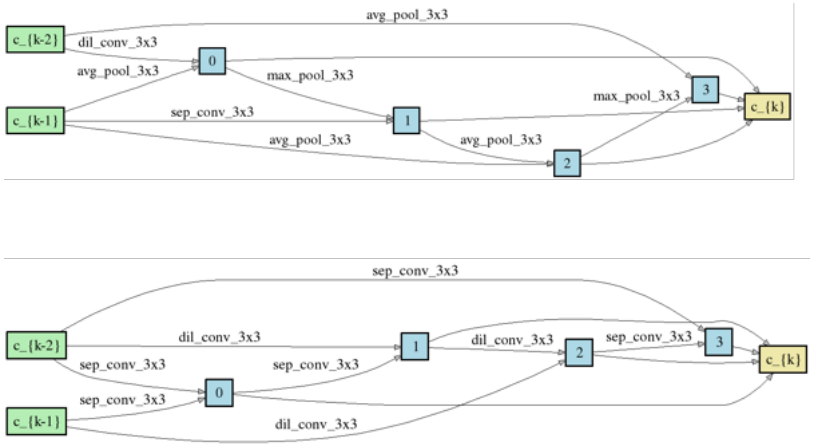


Figure A.33: Normal and Reduction cell in Epoch 38.

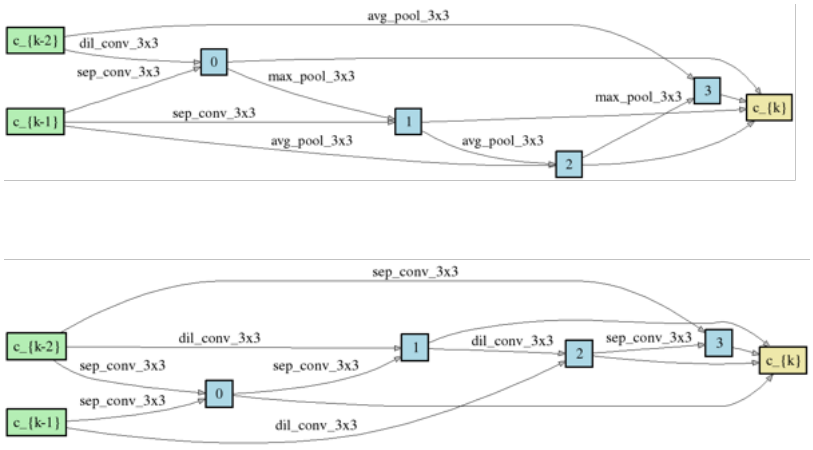


Figure A.34: Normal and Reduction cell in Epoch 39.

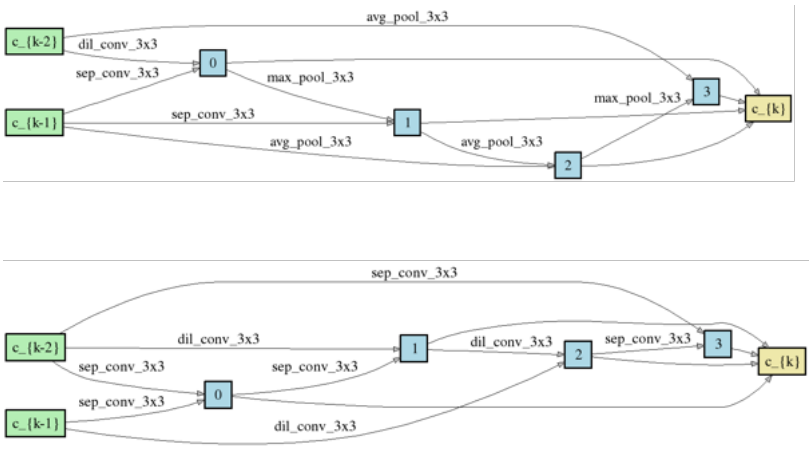


Figure A.35: Normal and Reduction cell in Epoch 40.

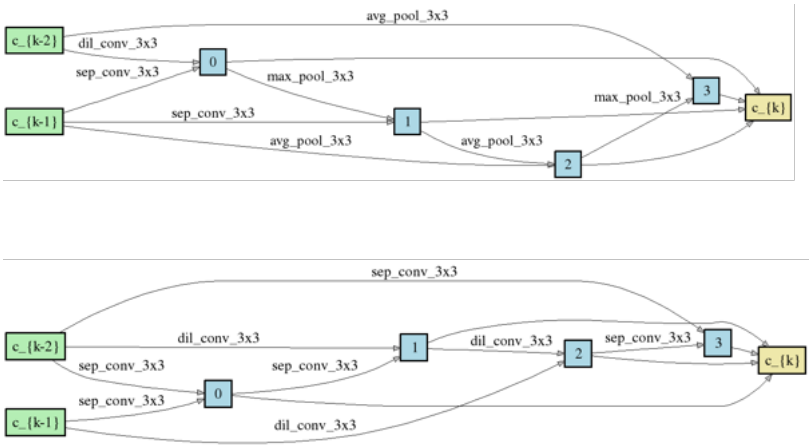


Figure A.36: Normal and Reduction cell in Epoch 41.

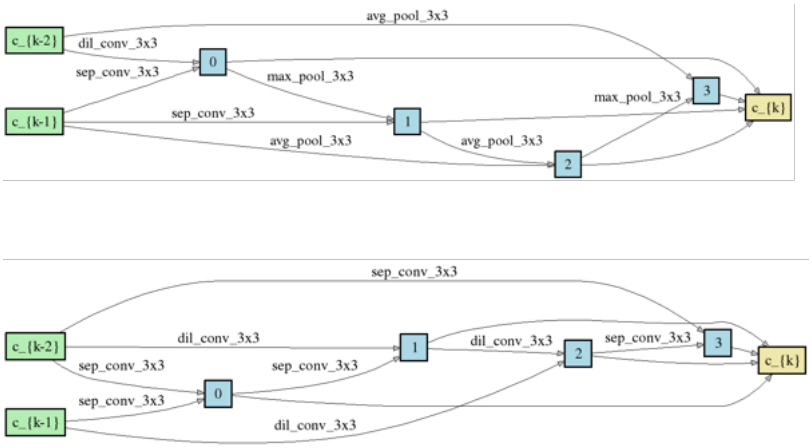


Figure A.37: Normal and Reduction cell in Epoch 42.

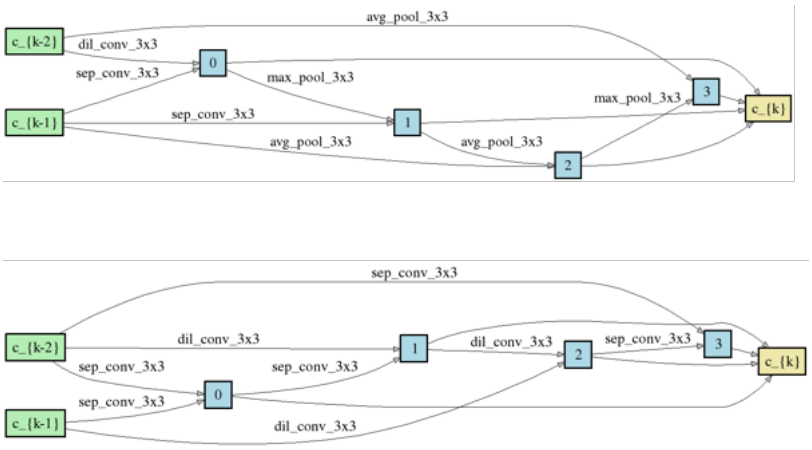


Figure A.38: Normal and Reduction cell in Epoch 43.

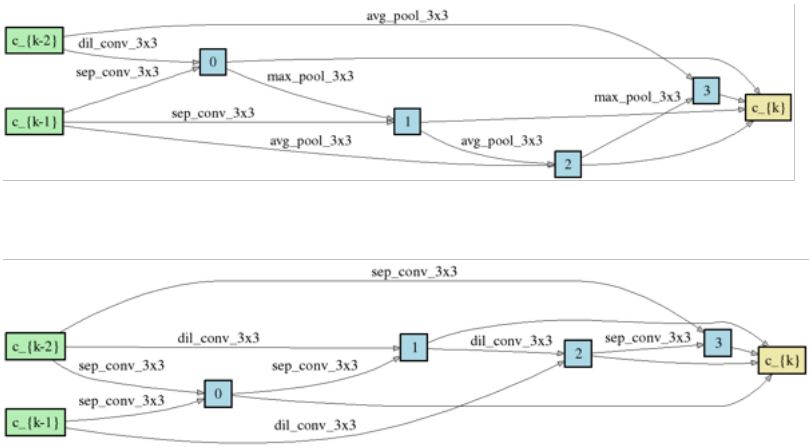


Figure A.39: Normal and Reduction cell in Epoch 44.

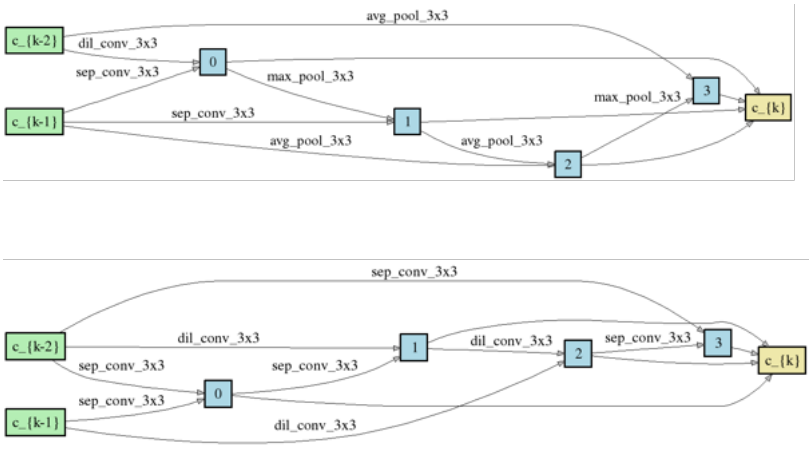


Figure A.40: Normal and Reduction cell in Epoch 45.

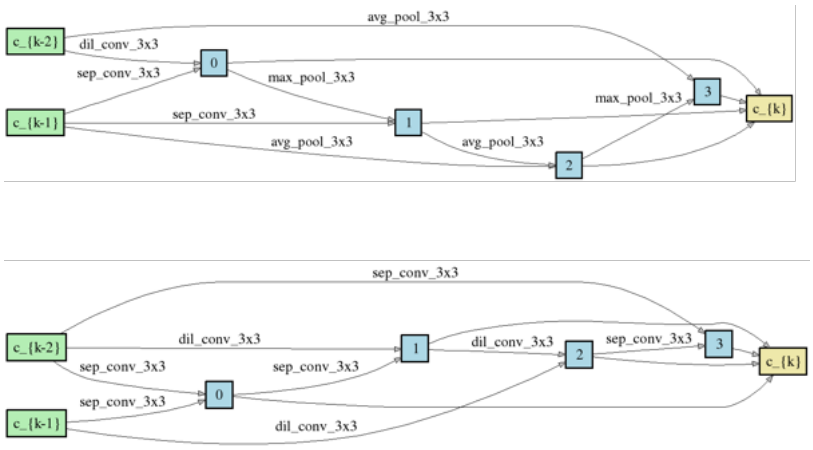


Figure A.41: Normal and Reduction cell in Epoch 46.

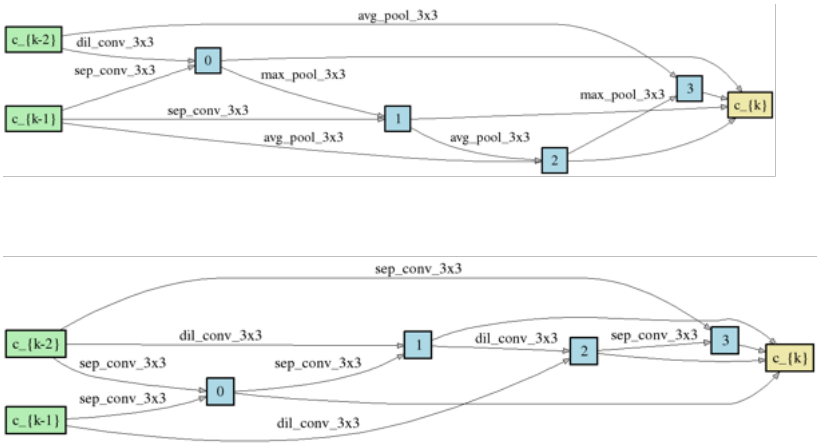


Figure A.42: Normal and Reduction cell in Epoch 47.

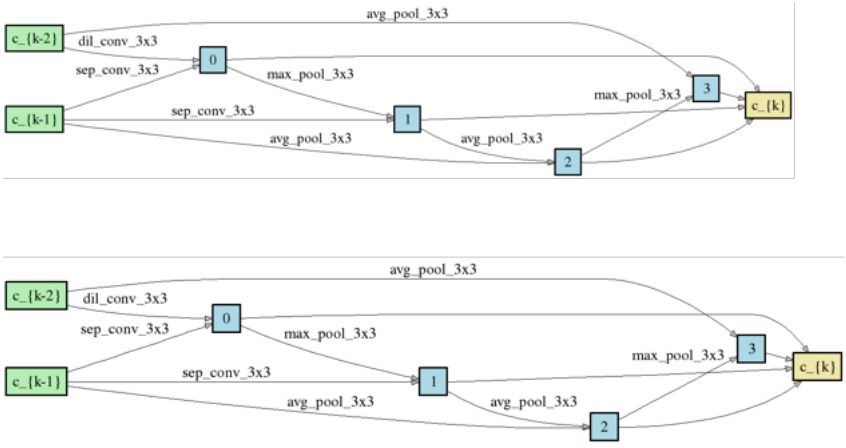


Figure A.43: Normal and Reduction cell in Epoch 48.

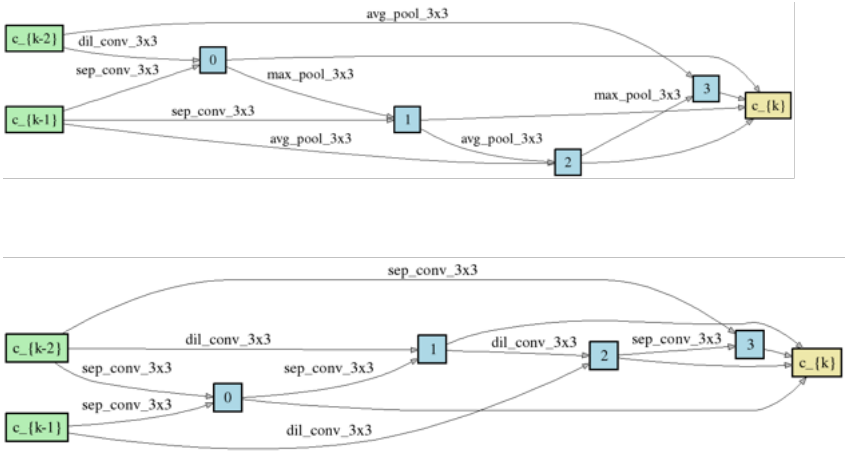


Figure A.44: Normal and Reduction cell in Epoch 49.

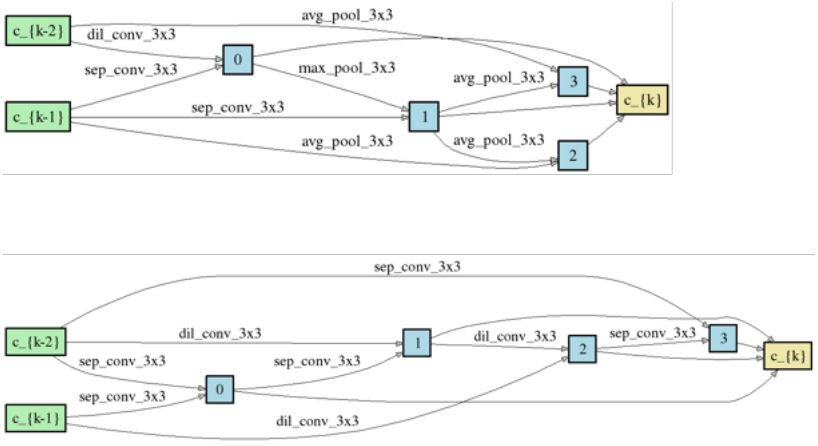


Figure A.45: Normal and Reduction cell in Epoch 50.

Table A.1: Accuracy and Loss for Seed 1

Epoch	trainaccdarts	trainlossdarts	validaccdarts	validlossdarts	trainacclstm	trainlosslstm	validacclstm	validlosslstm	trainacclstmembed	trainlosslstmembed	validacclstmembed	validlosslstmembed	trainaccenn	trainlosscnn	validaccenn	validlosscnn	trainaccennembed	trainlosscnnembed	validacccnnembed	validlosscnnembed	trainacchybrid	trainlosshybrid	validacchybrid	validlosshybrid
1	77.712872	0.460214	80.665959	0.424856	66.44	0.5928	85.92	0.3439	62.7	0.6409	68.51	0.6252	62.08	0.6466	63.47	0.6376	64.89	0.6214	63.24	0.6403	89.95	0.2532	93.35	0.1938
2	88.786545	0.29491	90.032548	0.269924	87.11	0.3242	91.45	0.2383	72.1	0.5503	82.96	0.4123	63.45	0.6373	63.66	0.6363	68.08	0.5944	58.25	0.7376	93.22	0.1953	93.58	0.183
3	90.776896	0.250309	91.841399	0.23712	90.53	0.2559	92.06	0.2223	83.69	0.3915	87.21	0.3274	63.56	0.6362	63.04	0.6401	68.92	0.5863	58.89	0.7166	93.78	0.1814	93.72	0.1817
4	91.884732	0.225843	90.816487	0.253311	91.54	0.2336	92.91	0.2056	85.53	0.3568	88.54	0.3013	63.75	0.6353	63.24	0.6397	69.57	0.5797	55.22	0.7581	94.1	0.172	94.25	0.1708
5	92.56383	0.208776	92.667586	0.219973	92.06	0.2217	93.37	0.193	86.71	0.3341	90.22	0.2653	63.81	0.6345	63.95	0.6332	69.88	0.5768	61.59	0.6662	94.27	0.1666	94.17	0.1724
6	92.982137	0.199887	93.162047	0.196124	92.34	0.2153	92.98	0.2021	87.86	0.3115	90.83	0.2515	63.81	0.6342	62.87	0.6368	70.2	0.573	61.23	0.6616	94.44	0.1619	94.19	0.1708
7	93.275787	0.193435	92.287352	0.215509	92.58	0.2085	93.42	0.1941	88.67	0.2952	91.62	0.2343	63.95	0.6335	62.51	0.6446	70.19	0.572	56.03	0.7484	94.59	0.1573	94.22	0.17
8	93.351416	0.191388	93.476561	0.188481	92.75	0.2058	93.47	0.1906	89.31	0.2812	91.94	0.2228	63.91	0.6333	63.73	0.6333	70.46	0.57	52.53	0.7443	94.75	0.1532	93.89	0.1761
9	93.407747	0.18937	93.608001	0.18646	92.94	0.2017	93.18	0.1956	89.8	0.2722	91.57	0.2338	63.95	0.6331	63.95	0.6321	70.6	0.5681	55.35	0.7335	94.8	0.1525	93.99	0.1746
10	93.517279	0.187357	92.939853	0.200162	93.07	0.1991	93.79	0.1816	90.13	0.2653	91.08	0.2519	64	0.6331	64.11	0.6322	70.65	0.5675	60.7	0.6685	94.98	0.1471	94.3	0.1669
11	93.502675	0.186978	93.626777	0.182038	93.12	0.1972	93.74	0.1812	90.3	0.2618	92.59	0.2102	64.06	0.6325	64.29	0.6293	70.74	0.5671	54.94	0.7156	95.11	0.1436	94.31	0.1686
12	93.531362	0.185827	93.633037	0.183308	93.2	0.1957	93.88	0.1832	90.57	0.2561	92.78	0.2075	64.28	0.6312	64.35	0.6288	70.89	0.5654	55.75	0.7249	95.26	0.1396	94.21	0.1728
13	93.544923	0.184125	93.582965	0.185274	93.26	0.1937	94.06	0.1763	90.69	0.253	93.02	0.2021	64.22	0.6308	64.19	0.6302	70.83	0.5651	54.59	0.7195	95.4	0.1356	94.24	0.1715
14	93.557962	0.183454	93.698756	0.182771	93.27	0.1932	93.99	0.1806	90.93	0.2473	93.17	0.2005	64.26	0.6307	64.44	0.6287	70.82	0.5647	58.63	0.6746	95.5	0.133	94.36	0.1742
15	93.576739	0.183814	93.665896	0.183343	93.37	0.1918	94.09	0.1733	91.05	0.2459	93.1	0.2023	64.31	0.6304	64.37	0.6284	70.99	0.564	59.05	0.6711	95.41	0.1332	94.16	0.1754
16	93.567351	0.183005	93.833324	0.180094	93.43	0.19	94.1	0.175	91.22	0.2422	93.23	0.1987	64.32	0.6304	64.04	0.6304	71.07	0.5634	55.29	0.7169	95.6	0.1291	94.11	0.1805
17	93.58352	0.182832	93.1464	0.201872	93.46	0.1892	94.11	0.1747	91.38	0.2391	93.08	0.2012	64.26	0.6302	64.28	0.6299	71.14	0.5618	56.46	0.6809	95.74	0.1247	94.1	0.1786
18	93.54753	0.184368	93.819242	0.180643	93.45	0.188	94.13	0.1727	91.42	0.2382	93.01	0.2026	64.4	0.6296	63.51	0.6342	71.07	0.5632	50.99	0.8046	95.77	0.1244	94.04	0.1846
19	93.616901	0.183187	93.049385	0.200715	93.47	0.1875	94.12	0.1748	91.55	0.2358	93.43	0.1949	64.44	0.6294	64.32	0.6309	71.1	0.562	52.29	0.7392	95.87	0.1205	94.05	0.1795
20	93.574653	0.183178	93.825501	0.178658	93.6	0.1862	93.94	0.1802	91.61	0.2345	93.39	0.1922	64.5	0.629	64.49	0.6288	71.15	0.5619	50.38	0.7678	96.05	0.1169	93.88	0.1848
21	93.565264	0.183699	93.755087	0.183505	93.62	0.1859	94.11	0.175	91.78	0.2311	93.5	0.1908	64.38	0.6291	64.52	0.6288	71.2	0.561	50.07	0.8142	96.01	0.1166	94.11	0.1841
22	93.58039	0.183039	93.572011	0.188627	93.53	0.1859	94.09	0.1763	91.75	0.2304	93.69	0.1883	64.43	0.629	64.6	0.6273	71.32	0.561	50.24	0.7646	96.08	0.114	94.07	0.1841
23	93.571002	0.183406	93.86462	0.180275	93.63	0.1843	94	0.1775	91.92	0.2275	93.73	0.1876	64.41	0.629	63.54	0.6362	71.21	0.5609	51.84	0.7435	96.22	0.1112	94.09	0.1939
24	93.508412	0.184069	93.847407	0.179184	93.58	0.1853	94.3	0.1724	91.96	0.2259	93.76	0.1856	64.57	0.6288	64.55	0.6286	71.33	0.5597	50.22	0.7744	95.77	0.1224	93.77	0.1852
25	93.52771	0.183934	93.817677	0.181637	93.68	0.1834	94.32	0.17	92.04	0.2248	93.78	0.183	64.56	0.6282	64.61	0.6263	71.39	0.5592	50.79	0.7742	96.13	0.1134	93.97	0.1849
26	93.536056	0.184044	93.742569	0.180702	93.64	0.183	94.33	0.1687	92.02	0.2232	93.74	0.1881	64.58	0.6283	64.73	0.6277	71.33	0.5595	53.17	0.7175	96.28	0.1088	93.94	0.1938
27	93.512063	0.183392	93.855231	0.178267	93.7	0.1825	94.22	0.1755	92.16	0.2214	93.83	0.1816	64.54	0.6283	64.52	0.6294	71.36	0.5597	51.36	0.7197	96.39	0.1055	94	0.1961
28	93.475552	0.184546	93.51881	0.188059	93.66	0.1835	94.19	0.1698	92.23	0.2193	93.52	0.1919	64.53	0.6281	64.38	0.632	71.5	0.5588	50.56	0.74	96.39	0.1055	93.81	0.196
29	93.549095	0.184001	93.98667	0.177365	93.77	0.1814	94.22	0.1698	92.26	0.2193	93.91	0.1808	64.59	0.6281	64.4	0.6333	71.39	0.5594	50.68	0.7941	96.22	0.1093	93.82	0.2024
30	93.545966	0.184531	93.789511	0.177862	93.73	0.1819	94.23	0.1698	92.25	0.2183	93.9	0.1823	64.57	0.6278	64.49	0.6295	71.47	0.5587	50.89	0.7822	96.15	0.1117	93.91	0.1914
31	93.485462	0.185168	93.902173	0.17687	93.77	0.1812	94.22	0.1718	92.32	0.2161	93.89	0.1782	64.48	0.628	64.55	0.6267	71.53	0.5586	50.34	0.7614	96.04	0.1165	93.49	0.1959
32	93.452081	0.185255	93.582965	0.185642	93.72	0.1807	94.35	0.1668	92.38	0.2159	93.99	0.1775	64.53	0.6282	64.73	0.6263	71.5	0.5584	50.49	0.8044	96.4	0.1046	93.84	0.2021
33	93.481811	0.185539	93.92408	0.177756	93.79	0.1802	94.33	0.1694	92.39	0.2146	93.72	0.1815	64.52	0.6282	64.45	0.6276	71.45	0.5583	49.97	0.8081	96.43	0.1045	93.85	0.2014
34	93.422351	0.185511	93.981975	0.175592	93.8	0.1799	94.35	0.1691	92.44	0.2142	93.54	0.1893	64.55	0.628	64.61	0.6277	71.41	0.5581	51.86	0.7104	96.3	0.1074	93.71	0.2006
35	93.486506	0.185256	94.074295	0.173941	93.81	0.18	94.28	0.1688	92.48	0.2135	94.05	0.1779	64.48	0.628	64.34	0.6304	71.51	0.5577	50.03	0.7873	96.61	0.0988	93.82	0.2011
36	93.434347	0.186546	93.920951	0.179565	93.81	0.1803	94.4	0.1684	92.51	0.2115	93.98	0.178	64.51	0.6278	64.29	0.6275	71.5	0.5573	50.89	0.8111	96.49	0.1023	93.92	0.2061
37	93.434347	0.185496	94.043001	0.175504	93.85	0.1787	94.39	0.1663	92.6	0.209	94.14	0.1756	64.51	0.6278	64.24	0.6296	71.5	0.5577	50.01	0.7967	96.61	0.0984	93.51	0.2094
38	93.413484	0.186292	94.105591	0.175895	93.87	0.1789	94.35	0.1691	92.54	0.2094	94.04	0.1759	64.5	0.6278	64.61	0.628	71.57	0.5574	50.04	0.7949	96.12	0.1128	93.8	0.2086
39	93.41192	0.186683	93.467173	0.191193	93.83	0.1785	94.32	0.1672	92.57	0.2102	94.14	0.1739	64.67	0.6274	64.45	0.628	71.56	0.5572	50	0.7969	96.43	0.104	93.9	0.1989
40	93.399402	0.186841	93.773864	0.183868	93.82	0.1788	94.32	0.1657	92.65	0.208	94.13	0.175	64.67	0.6273	64.15	0.6318	71.55	0.5574	49.99	0.7618	96.55	0.1003	93.7	0.2001

Table A.2: Accuracy and Loss for Seed 2

Epoch	trainaccdarts	trainlossdarts	validaccdarts	validlossdarts	trainacclstm	trainlosslstm	validacclstm	validlosslstm	trainacclstmembed	trainlosslstmembed	validacclstmembed	validlosslstmembed	trainaccenn	trainlosscnn	validaccenn	validlosscnn	trainaccennembed	trainlosscnnembed	validaccennembed	validlosscnnembed	trainacchybrid	trainlosshybrid	validacchybrid	validlosshybrid
1	79.776765	0.434363	86.831071	0.329752	67.01	0.5968	78.7	0.475	70.76	0.5637	82.66	0.4441	61.79	0.6488	63.37	0.6366	64.46	0.6245	64.05	0.6346	87.72	0.2877	92.98	0.1976
2	89.515714	0.279448	91.204545	0.242379	82.98	0.4014	88.77	0.2934	79.62	0.4556	85	0.3681	63.04	0.6405	63.45	0.6365	68.1	0.594	63.84	0.6364	93.15	0.1958	93.35	0.1933
3	91.393404	0.238275	92.046381	0.222877	88.57	0.2977	91.38	0.2388	84.36	0.3795	87.54	0.315	63.26	0.6391	63.55	0.6355	69.09	0.5847	62.94	0.6431	93.53	0.1855	94.05	0.1741
4	92.186206	0.219767	92.054204	0.222971	90.21	0.2623	92.39	0.2156	86.36	0.3398	89.16	0.2865	63.42	0.6365	63.62	0.6339	69.49	0.5804	58.7	0.6837	93.97	0.1744	94.05	0.1742
5	92.575305	0.209609	93.126058	0.200149	91.19	0.2409	91.8	0.2262	88.01	0.3087	90.89	0.2507	63.51	0.6359	63.93	0.6325	69.73	0.5768	63.38	0.6384	94.22	0.1683	94.24	0.1698
6	92.812624	0.203105	93.244979	0.194026	90.13	0.2552	92.04	0.2239	89.23	0.285	91.59	0.2331	63.67	0.6347	63.97	0.6312	69.99	0.5742	61.45	0.6545	94.37	0.1641	94.27	0.169
7	93.125051	0.197336	91.625463	0.234251	91.97	0.2241	92.86	0.2115	89.97	0.268	92.11	0.2217	63.83	0.6345	64.07	0.6337	70.1	0.5731	60.92	0.6603	94.59	0.1597	94.11	0.1713
8	93.249186	0.192698	91.633287	0.234126	92.21	0.2194	93.37	0.1926	90.48	0.2575	92.46	0.2127	63.65	0.6344	64.09	0.6319	70.31	0.5718	61.74	0.6506	94.66	0.1559	94.33	0.1698
9	93.313341	0.190039	89.445767	0.276907	92.27	0.217	93.18	0.1978	90.82	0.2499	92.39	0.2128	63.71	0.6339	63.86	0.6316	70.42	0.5698	60.86	0.657	94.83	0.1519	94.14	0.1685
10	93.502153	0.187217	93.439007	0.188346	92.71	0.2088	93.45	0.1977	91.08	0.2444	92.81	0.2062	63.77	0.6338	63.73	0.6323	70.62	0.5683	58.56	0.6795	94.93	0.148	94.39	0.1657
11	93.513106	0.185945	93.417101	0.192951	90.25	0.2489	93.2	0.201	91.29	0.2397	93	0.2014	63.71	0.6336	64.02	0.6318	70.66	0.5675	61.93	0.6474	95.07	0.1444	94.3	0.1665
12	93.525624	0.184337	93.767605	0.17937	92.78	0.2067	93.82	0.1849	91.46	0.2359	93	0.2008	63.85	0.633	63.89	0.6308	70.75	0.5671	56.38	0.6903	95.25	0.1405	94.24	0.1729
13	93.634113	0.183069	93.85836	0.180044	92.96	0.2015	93.84	0.1806	91.6	0.2318	93.2	0.1936	63.77	0.6332	64.18	0.6301	70.76	0.5663	60.53	0.6576	95.34	0.1373	94.42	0.1698
14	93.565786	0.182644	91.260877	0.245491	93.08	0.1988	93.55	0.1894	91.73	0.2293	93.04	0.1997	63.91	0.633	62.41	0.6424	70.85	0.5653	56.76	0.6992	95.46	0.1339	94.34	0.1678
15	93.63307	0.182283	94.033613	0.175792	93.19	0.1966	93.69	0.183	91.92	0.2261	93.39	0.191	63.83	0.6329	64.05	0.631	70.78	0.5647	53.86	0.7252	95.55	0.1313	94.3	0.1746
16	93.641937	0.181746	93.376418	0.189843	93.26	0.1945	93.92	0.1801	91.96	0.2239	93.33	0.1952	63.85	0.633	63.96	0.6314	70.95	0.564	60.08	0.658	95.7	0.1282	94.25	0.1763
17	93.658105	0.181036	92.681669	0.208683	93.28	0.1938	93.96	0.1762	91.96	0.2224	93.58	0.187	63.85	0.6326	64.08	0.6308	71.02	0.5636	62.58	0.6451	95.75	0.1257	94.1	0.1751
18	93.705048	0.180149	94.000752	0.177666	93.41	0.1914	93.92	0.1812	92.08	0.2226	93.48	0.189	63.9	0.6327	64.07	0.6304	71.04	0.5638	60.95	0.6531	95.83	0.1241	94.23	0.1761
19	93.688357	0.180635	93.856796	0.178209	93.37	0.1914	94.02	0.1804	92.1	0.2206	93.67	0.1844	63.81	0.6326	64.06	0.63	70.97	0.5624	58.76	0.6623	95.94	0.1206	94.22	0.1809
20	93.645588	0.18131	94.091508	0.172163	93.41	0.1907	93.83	0.1838	92.23	0.2183	93.55	0.1876	63.85	0.6325	64	0.6311	70.94	0.5632	57.97	0.67	96	0.1188	93.97	0.1814
21	93.629419	0.180607	93.903738	0.176944	93.49	0.1885	94.23	0.1725	92.32	0.2169	93.68	0.1839	63.95	0.6322	63.83	0.6328	70.99	0.5628	55.42	0.7139	96.09	0.1164	94.16	0.1823
22	93.594473	0.180889	94.132192	0.173846	93.45	0.1896	94.13	0.1749	92.35	0.2159	93.3	0.1983	63.86	0.6324	64.12	0.6308	71.09	0.5622	61.7	0.6512	96.14	0.1146	94.19	0.1834
23	93.596037	0.180288	93.917821	0.177877	93.56	0.1879	93.95	0.1832	92.42	0.2146	93.78	0.1828	63.9	0.6323	63.97	0.6317	71.14	0.5614	59.73	0.6641	96.2	0.113	94.23	0.1847
24	93.554832	0.181148	93.70345	0.185522	93.51	0.1875	94.07	0.1752	92.4	0.2135	93.15	0.2046	63.88	0.6321	63.98	0.6308	71.22	0.5623	56.13	0.6981	96.29	0.1109	94.08	0.186
25	93.599167	0.181677	94.176004	0.172502	93.59	0.1866	94.2	0.1727	92.46	0.2127	93.55	0.1863	63.95	0.632	64.03	0.6293	71.16	0.5609	57.78	0.67	96.33	0.109	94.06	0.1857
26	93.640893	0.180785	93.952245	0.180151	93.6	0.1851	94.27	0.1714	92.52	0.2116	93.84	0.1799	63.96	0.632	64.1	0.63	71.11	0.5614	52.09	0.7301	96.35	0.1083	94.07	0.1876
27	93.613771	0.181739	94.149404	0.172689	93.57	0.186	94.2	0.1721	92.53	0.2119	93.54	0.1914	63.91	0.6317	64.12	0.6332	71.15	0.5614	54.68	0.7035	96.41	0.1074	93.93	0.1936
28	93.581955	0.180915	93.989799	0.17453	93.63	0.1851	93.94	0.1801	92.52	0.2111	93.86	0.1804	63.9	0.6322	64.12	0.63	71.18	0.5605	54.37	0.6866	96.43	0.1059	94.06	0.1952
29	93.612728	0.18109	94.127497	0.174315	93.4	0.1911	94.26	0.1711	92.61	0.2097	93.83	0.1823	63.9	0.632	63.98	0.6325	71.14	0.561	55.29	0.7055	96.47	0.1063	93.97	0.1952
30	93.627332	0.181904	94.021094	0.17413	93.46	0.1901	94	0.1836	92.64	0.2088	93.78	0.1806	63.84	0.6319	64.02	0.6306	71.15	0.5612	54.73	0.7136	96.5	0.104	93.89	0.1939
31	93.524059	0.182275	93.090068	0.202759	93.69	0.1843	94.03	0.1791	92.63	0.2078	93.61	0.1854	63.95	0.6319	63.98	0.6299	71.2	0.5606	56.39	0.7007	96.55	0.103	94.04	0.1984
32	93.555876	0.182117	92.005697	0.227999	92.68	0.2042	94.09	0.1781	92.65	0.2082	93.85	0.1785	63.91	0.6319	63.49	0.6333	71.2	0.5608	53.58	0.7164	96.56	0.1029	93.99	0.1938
33	93.563178	0.182907	93.603306	0.187391	93.65	0.1846	94.19	0.1748	92.7	0.2072	93.77	0.1835	64.02	0.6319	63.71	0.6331	71.31	0.5602	52.23	0.7356	96.62	0.1014	93.74	0.204
34	93.496937	0.183917	93.903738	0.182632	93.69	0.1839	94.19	0.1714	92.73	0.2067	93.94	0.1765	63.92	0.6318	64.01	0.6309	71.29	0.5599	56.35	0.6881	96.6	0.1007	94.01	0.1996
35	93.505804	0.183069	94.287102	0.170088	93.51	0.1884	94.13	0.1769	92.69	0.2063	93.99	0.1773	64.03	0.6315	64.02	0.6292	71.27	0.5601	54.15	0.7312	96.65	0.0993	93.94	0.1977
36	93.488592	0.183934	93.121364	0.201011	93.7	0.1841	93.72	0.1902	92.76	0.2051	93.78	0.1854	63.92	0.6318	64.15	0.6295	71.29	0.5591	58.88	0.6686	96.51	0.1028	93.99	0.1974
37	93.470858	0.184805	92.589349	0.217748	93.59	0.1874	94.31	0.1715	92.79	0.2047	94.06	0.1757	63.9	0.6314	63.81	0.6323	71.32	0.5593	56.12	0.6995	96.51	0.1028	93.99	0.1974
38	93.444779	0.185408	93.326346	0.190604	93.68	0.1836	94.29	0.1721	92.77	0.2053	93.97	0.1779	63.96	0.6314	64.03	0.6297	71.29	0.5592	55.56	0.7096	96.51	0.1028	93.99	0.1974
39	93.446866	0.18545	93.98354	0.177732	93.67	0.1827	94.25	0.1689	92.85	0.2039	93.78	0.1819	63.99	0.6313	64	0.6322	71.3	0.5599	52.64	0.7477	96.51	0.1028	93.99	0.1974
40	93.464078	0.184911	94.258936	0.171081	93.79	0.1813	94.33	0.1715	92.82	0.2048	93.65	0.1829	63.97	0.6315	63.56	0.635	71.38	0.5585	56.55	0.6921	96.51	0.1028	93.99	0.1974

Table A.3: Accuracy and Loss for Seed 3

Epoch	trainaccdarts	trainlossdarts	validaccdarts	validlossdarts	trainacclstm	trainlosslstm	validacclstm	validlosslstm	trainacclstmembed	trainlosslstmembed	validacclstmembed	validlosslstmembed	trainaccenn	trainlosscnn	validaccenn	validlosscnn	trainaccennembed	trainlosscnnembed	validaccennembed	validlosscnnembed	trainacchybrid	trainlosshybrid	validacchybrid	validlosshybrid
1	77.665929	0.464988	86.000189	0.367108	62.07	0.6423	70.16	0.593	61.17	0.6525	50.01	0.6894	61.49	0.6507	62.49	0.6439	64.4	0.6254	62.03	0.6617	90.2	0.2503	92.58	0.2133
2	89.094799	0.289239	90.694437	0.25528	73.62	0.5333	85.03	0.3709	74.1	0.5098	85.99	0.3486	62.73	0.6434	63.15	0.6393	67.87	0.5962	63.61	0.6412	93.43	0.1887	93.94	0.1779
3	91.453385	0.235041	91.99005	0.222028	85.83	0.3452	65.51	0.6233	85.36	0.3609	88.03	0.3078	62.91	0.6416	63.29	0.6383	68.95	0.5866	63.63	0.6423	93.87	0.1786	94.06	0.1728
4	92.332248	0.214334	92.780248	0.206094	81.1	0.4241	88.43	0.2999	86.93	0.328	89.5	0.2814	62.9	0.6407	63.03	0.6402	69.37	0.5817	62.78	0.6423	94.14	0.1712	93.39	0.1913
5	92.86739	0.201333	93.05095	0.198128	89.61	0.2735	92.2	0.216	88.18	0.3041	90.67	0.2521	62.91	0.6403	62.96	0.6398	69.65	0.578	58.98	0.6917	94.32	0.1658	94.25	0.1687
6	93.189726	0.194246	92.76773	0.204066	91.4	0.2361	92.45	0.2187	89.09	0.2855	91.91	0.2256	63.03	0.6399	61.83	0.6441	69.92	0.5756	58.59	0.6935	94.51	0.1606	93.96	0.1751
7	93.351938	0.189542	92.730176	0.204869	92.05	0.2221	92.78	0.2026	89.74	0.2716	91.58	0.2302	63.1	0.639	63.3	0.6363	70.21	0.5733	59.69	0.6695	94.63	0.157	94.01	0.1726
8	93.47138	0.186965	93.343558	0.192174	92.43	0.2129	93.52	0.1858	90.19	0.2623	91.91	0.224	63.45	0.638	63.28	0.6399	70.2	0.5727	56.6	0.6982	94.76	0.1524	94.26	0.1679
9	93.509977	0.185631	93.296616	0.19126	92.65	0.2078	93.62	0.1875	90.41	0.2566	91.97	0.2224	63.52	0.6373	63.73	0.634	70.38	0.5712	54.96	0.7597	94.94	0.1483	94.2	0.1685
10	93.548052	0.185215	93.517245	0.185951	92.8	0.2053	93.31	0.1891	90.73	0.2489	93.05	0.1977	63.64	0.6361	63.9	0.6336	70.32	0.5699	58.28	0.6772	95.11	0.1441	94.15	0.1709
11	93.589779	0.183455	92.939853	0.208091	92.93	0.2011	93.64	0.1877	90.05	0.2588	92.98	0.201	63.69	0.6358	63.69	0.637	70.55	0.5692	55.98	0.7121	95.18	0.1407	94.12	0.1699
12	93.660713	0.182285	93.642425	0.188743	92.98	0.2002	93.94	0.1794	91.07	0.2432	93.21	0.1955	63.64	0.636	64.07	0.6325	70.75	0.5676	56.13	0.6997	95.31	0.1368	94.15	0.1719
13	93.639329	0.181813	93.507857	0.187413	93.08	0.1979	93.69	0.1823	91.3	0.2374	93.35	0.1917	63.63	0.6358	63.51	0.6347	70.74	0.567	55.07	0.6982	95.45	0.1338	94.06	0.1795
14	93.633591	0.181619	93.784817	0.181302	93.16	0.1956	93.65	0.1831	90.22	0.257	93.44	0.1907	63.68	0.6353	63.82	0.6352	70.68	0.5667	56.46	0.6903	95.56	0.1303	94.02	0.1761
15	93.683141	0.18138	93.446831	0.189376	93.24	0.1944	93.94	0.1801	91.45	0.2339	93.39	0.1895	63.82	0.6351	63.68	0.6349	70.77	0.5662	51.95	0.8013	95.63	0.1273	94.09	0.1748
16	93.771289	0.179593	93.697191	0.182164	93.26	0.1935	93.99	0.1758	91.52	0.2324	93.48	0.1863	63.78	0.635	63.14	0.6385	70.84	0.5655	55.2	0.7224	95.76	0.1234	94.1	0.1789
17	93.706613	0.180161	93.446831	0.188805	93.25	0.1938	94.16	0.1726	91.63	0.2304	93.48	0.1879	63.74	0.6349	64.09	0.6319	70.9	0.5646	52.85	0.7415	95.84	0.1215	94.16	0.1803
18	93.6628	0.180545	93.756652	0.181121	93.3	0.1926	94.1	0.1744	91.69	0.2278	93.59	0.1891	63.76	0.6349	63.83	0.6343	70.87	0.5653	54.6	0.7374	95.99	0.118	94.18	0.1784
19	93.683663	0.18015	93.82863	0.179796	93.35	0.1909	93.46	0.1921	91.82	0.225	93.81	0.1826	63.82	0.6346	64.05	0.632	71.07	0.5638	53.2	0.7153	95.97	0.1167	94.19	0.1836
20	93.69253	0.180669	93.573576	0.186942	93.44	0.1891	94.12	0.1741	91.95	0.2228	93.3	0.1912	63.8	0.6346	63.92	0.6316	70.97	0.5639	52.91	0.7089	96.05	0.1153	93.94	0.1863
21	93.726954	0.179063	93.521939	0.189523	93.39	0.1905	93.73	0.1821	91.79	0.2263	93.69	0.1834	63.8	0.6343	63.47	0.6349	71.02	0.5635	51.95	0.7217	96.1	0.1128	93.86	0.1985
22	93.696703	0.179963	93.816112	0.17851	93.44	0.1889	94.25	0.1714	91.79	0.227	93.68	0.1817	63.86	0.6343	64.01	0.633	71.07	0.5628	50.13	0.8221	96.19	0.1105	93.93	0.1894
23	93.645066	0.180329	93.805159	0.17945	93.47	0.189	94.01	0.1797	92.05	0.2208	93.75	0.1815	63.83	0.6343	64	0.6335	71.1	0.5627	50.4	0.8134	96.29	0.1074	94.05	0.1914
24	93.734778	0.178995	92.548665	0.212516	93.48	0.1875	94.06	0.1764	92.11	0.2198	93.74	0.1821	63.89	0.634	63.54	0.6368	71.02	0.5629	53.03	0.7016	96.32	0.1065	93.7	0.1986
25	93.718609	0.17953	93.903738	0.179146	93.45	0.1889	94.13	0.1807	92.18	0.2174	93.84	0.1803	63.79	0.6344	64.07	0.6348	71.16	0.5613	50.12	0.8188	96.39	0.105	93.92	0.194
26	93.609077	0.180922	93.903738	0.178936	93.51	0.1878	94.23	0.1723	92.21	0.2165	93.62	0.1865	63.86	0.6343	64.11	0.6319	71.06	0.5624	51.91	0.7525	96.44	0.1022	94.01	0.1986
27	93.730084	0.179658	93.874008	0.176043	93.45	0.1889	94.02	0.1737	92.27	0.2153	93.96	0.1774	63.86	0.6339	64.23	0.6305	71.16	0.5627	50.75	0.7575	96.53	0.1001	93.78	0.2025
28	93.664886	0.179458	93.76604	0.178365	93.55	0.1863	94.26	0.1696	92.4	0.2135	93.95	0.1799	64.08	0.6324	64.25	0.6303	71.22	0.5618	50.25	0.783	96.53	0.1002	93.86	0.2031
29	93.604383	0.180977	93.315392	0.200196	93.66	0.1852	94.09	0.1745	92.39	0.2138	93.78	0.1829	64.09	0.6318	64.35	0.6293	71.23	0.5609	50.6	0.7963	96.53	0.0991	93.85	0.1982
30	93.676883	0.180013	93.617389	0.185909	93.61	0.1853	94.02	0.1776	92.33	0.214	93.97	0.1759	64.13	0.6313	64.09	0.6304	71.07	0.5624	50.18	0.767	96.61	0.0983	93.91	0.2053
31	93.597081	0.181232	94.01953	0.177102	93.63	0.1844	94.34	0.1687	92.46	0.2113	94.01	0.1757	64.05	0.6317	64.41	0.6292	71.2	0.5607	51.65	0.7554	96.62	0.0967	93.79	0.2134
32	93.608034	0.180958	93.930339	0.176363	93.58	0.1849	93.82	0.1836	92.45	0.2109	94.03	0.174	64.23	0.6312	64.43	0.6293	71.2	0.5614	51.82	0.7469	96.62	0.0969	93.87	0.2025
33	93.601775	0.180969	93.836454	0.179205	93.67	0.184	94.28	0.1689	92.5	0.2099	93.97	0.1757	64.19	0.6312	64.43	0.6297	71.29	0.5604	53.18	0.7254	96.64	0.0958	93.72	0.2043
34	93.656019	0.180537	93.927209	0.175063	93.72	0.183	94.22	0.1711	92.31	0.2148	93.98	0.1752	64.16	0.6313	64.18	0.6313	71.24	0.5614	50.77	0.7852	96.73	0.0939	93.69	0.2108
35	93.579347	0.182979	94.060213	0.177135	93.69	0.1833	94.05	0.1738	92.58	0.209	93.96	0.1759	64.15	0.6308	64.55	0.6279	71.22	0.5607	50.06	0.8314	96.74	0.0932	93.69	0.2108
36	93.621073	0.181511	93.899044	0.180584	93.7	0.1828	93.86	0.1815	92.6	0.2085	93.91	0.1755	64.23	0.6309	64.12	0.6315	71.26	0.5603	50.07	0.7885	96.7	0.0943	93.82	0.2076
37	93.532926	0.182858	93.476562	0.187691	93.62	0.1845	94.26	0.1713	92.6	0.2078	94.08	0.1726	64.18	0.6307	64.33	0.6296	71.36	0.5602	51.62	0.7545	96.74	0.0936	93.71	0.2101
38	93.479203	0.183714	93.79577	0.179805	93.71	0.1817	94.36	0.1683	92.6	0.2068	94.11	0.1738	64.17	0.6311	64.5	0.6288	71.32	0.5599	50.82	0.7285	96.82	0.0909	93.76	0.2104
39	93.54075	0.183866	93.85836	0.178448	93.73	0.1822	94.35	0.1673	92.21	0.216	94.03	0.1739	64.11	0.6313	64.37	0.6299	71.28	0.5601	50.73	0.7733	96.88	0.089	93.67	0.2156
40	93.480247	0.184368	93.996058	0.174314	93.69	0.1818	94.18	0.1704	92.32	0.2143	93.89	0.1765	64.25	0.6303	64.15	0.6292	71.28	0.5599	50.35	0.7869	96.87	0.0892	93.44	0.2229

Table A.4: Accuracy and Loss for Seed 4

Epoch	trainaccdarts	trainlossdarts	validaccdarts	validlossdarts	trainacclstm	trainlosslstm	validacclstm	validlosslstm	trainacclstmembed	trainlosslstmembed	validacclstmembed	validlosslstmembed	trainaccenn	trainlosscnn	validaccenn	validlosscnn	trainaccennembed	trainlosscnnembed	validaccennembed	validlosscnnembed	trainacchybrid	trainlosshybrid	validacchybrid	validlosshybrid
1	79.398098	0.43986	87.594669	0.321737	61.41	0.652	68.01	0.5944	69.25	0.5835	79.27	0.502	62.02	0.6466	64.07	0.6322	64.48	0.624	63.84	0.636	89.19	0.2651	92.57	0.2108
2	88.954494	0.29122	89.583466	0.276939	68.31	0.5828	61.99	0.65	82.29	0.42	85.36	0.3545	63.73	0.6357	64.14	0.6309	67.84	0.5962	61.93	0.6665	93.33	0.1914	93.54	0.1909
3	90.706482	0.252465	90.528574	0.257058	74.09	0.5259	85.49	0.3541	84.82	0.3719	87.74	0.3154	63.87	0.6339	63.96	0.6328	68.68	0.5878	62.79	0.6424	93.8	0.1803	93.92	0.1814
4	91.614553	0.229752	91.947801	0.221285	86.2	0.3382	88.23	0.3036	86.4	0.3426	89.31	0.2841	63.97	0.6327	63.98	0.6323	69.26	0.5819	62.71	0.6446	94.11	0.172	94.18	0.1714
5	92.38232	0.212817	92.673845	0.207952	90.17	0.262	92.4	0.2133	87.81	0.3151	90.9	0.2494	64.1	0.6324	64.33	0.6298	69.54	0.579	62.26	0.6511	94.34	0.1662	94.04	0.1726
6	92.868433	0.201868	91.972838	0.221168	91.36	0.236	92.95	0.2006	88.96	0.2904	91	0.2479	64.12	0.6317	63.84	0.6337	69.82	0.5771	60.44	0.6673	94.42	0.1633	93.91	0.176
7	93.08854	0.19682	92.567442	0.218275	91.91	0.2249	93	0.1974	89.61	0.2763	91.63	0.2311	64.17	0.6313	64.42	0.6306	70.05	0.5734	62.31	0.6435	94.62	0.1571	94.21	0.1701
8	93.264312	0.192817	93.082244	0.203533	92.29	0.2163	93.08	0.1954	90.02	0.2674	91.59	0.2378	64.27	0.6313	64.44	0.6287	70.19	0.5723	63.24	0.6388	94.78	0.1529	94.17	0.1704
9	93.319078	0.189962	93.439008	0.188641	92.45	0.2112	93.61	0.1877	90.44	0.26	92.28	0.2213	64.23	0.6308	64.3	0.63	70.23	0.5718	63.06	0.6404	94.91	0.1485	94.13	0.1734
10	93.435391	0.187528	93.528198	0.186756	92.75	0.2055	93.81	0.1816	90.63	0.255	92.51	0.2179	64.26	0.6307	64.45	0.6279	70.39	0.5706	62.68	0.642	95.02	0.1445	94.22	0.1709
11	93.489635	0.187097	93.751957	0.184256	92.89	0.2033	93.75	0.1809	90.86	0.2492	92.45	0.2134	64.25	0.6306	64.44	0.6287	70.55	0.5695	56.39	0.6966	95.23	0.1404	94.19	0.1736
12	93.54388	0.184175	93.784817	0.180942	92.99	0.1994	93.87	0.1779	91.04	0.2461	92.64	0.2109	64.23	0.6304	64.5	0.628	70.6	0.569	59.95	0.6609	95.3	0.1368	94.29	0.172
13	93.569437	0.184529	93.715968	0.183264	93.1	0.1967	93.94	0.1759	91.18	0.2422	92.59	0.2146	64.25	0.6304	64.29	0.6298	70.64	0.5677	64.34	0.632	95.44	0.1341	94.07	0.1783
14	93.628897	0.183188	93.000878	0.201318	93.17	0.1953	94.04	0.1766	91.29	0.2396	92.39	0.2212	64.24	0.6305	64.37	0.6295	70.51	0.5682	59.11	0.6672	95.56	0.1303	93.92	0.1797
15	93.606991	0.183273	93.11041	0.195221	93.19	0.1947	94.08	0.1746	91.37	0.237	93.14	0.1992	64.32	0.6297	64.36	0.63	70.75	0.5666	60.5	0.6582	95.7	0.1267	93.92	0.1821
16	93.65289	0.181176	93.500033	0.187552	93.34	0.1914	94.01	0.1758	91.52	0.2348	93.32	0.1959	64.4	0.6296	64.53	0.6276	70.68	0.567	59.85	0.66	95.83	0.1235	94.14	0.1808
17	93.703483	0.180739	93.902173	0.177116	93.31	0.1918	94.11	0.172	91.65	0.2317	93.4	0.1925	64.54	0.6288	64.55	0.628	70.67	0.5668	59.02	0.6727	95.9	0.1204	93.86	0.185
18	93.743123	0.180103	93.016525	0.19886	93.31	0.1917	94.12	0.1724	91.8	0.2288	93.32	0.1966	64.49	0.6289	64.56	0.6285	70.88	0.5651	60.53	0.6559	95.92	0.1191	93.9	0.1862
19	93.690444	0.180773	93.723792	0.179885	93.38	0.1893	94.2	0.1718	91.86	0.227	93.59	0.1895	64.51	0.6287	64.12	0.6325	70.84	0.565	59.1	0.6743	96.01	0.1164	94	0.1913
20	93.699832	0.180334	93.816112	0.179717	93.38	0.1902	94.16	0.1717	91.98	0.2257	93.51	0.1898	64.48	0.6282	64.37	0.6297	70.8	0.5652	60.27	0.658	96.13	0.114	93.81	0.1924
21	93.676361	0.18049	93.972587	0.174794	93.5	0.1871	94	0.1746	92.02	0.2242	93.61	0.1894	64.44	0.6285	64.46	0.6285	70.95	0.5641	59.51	0.6623	96.17	0.1119	93.94	0.189
22	93.701918	0.180143	93.434313	0.190625	90.78	0.2379	93.89	0.1791	92.06	0.2226	93.06	0.2066	64.54	0.6283	64.44	0.6319	70.98	0.5631	63.4	0.638	96.24	0.1106	93.8	0.1966
23	93.657584	0.180796	93.514116	0.189194	93.46	0.1879	94.25	0.1693	92.1	0.222	93.48	0.1889	64.48	0.6286	64.64	0.6279	70.95	0.5633	62.62	0.6437	96.28	0.1091	93.93	0.1879
24	93.618465	0.181068	93.654943	0.183882	93.46	0.1871	94.38	0.17	92.17	0.2185	93.68	0.1841	64.47	0.6283	63.98	0.6333	70.86	0.5643	58.56	0.6715	96.26	0.1094	93.99	0.1936
25	93.687314	0.180063	93.582965	0.184263	93.61	0.1851	94.21	0.1711	92.29	0.2169	93.68	0.1865	64.53	0.6278	64.54	0.6273	71.08	0.5632	60.63	0.6622	96.32	0.1074	93.72	0.1988
26	93.662278	0.18116	93.525069	0.186166	93.54	0.1855	94.22	0.173	92.21	0.2198	93.64	0.1848	64.52	0.628	64.61	0.629	71.02	0.5628	59.68	0.664	96.42	0.1056	93.72	0.197
27	93.694616	0.180113	93.874008	0.177589	93.64	0.1841	94.33	0.1687	92.38	0.2146	93.78	0.1824	64.51	0.6278	64.13	0.6298	71.08	0.5628	62.38	0.648	96.43	0.1053	93.86	0.2045
28	93.621073	0.181757	93.70345	0.181533	93.66	0.1833	94.17	0.1746	92.35	0.2161	93.72	0.1878	64.52	0.6279	64.58	0.6282	71.08	0.5628	58.73	0.6658	96.45	0.104	93.8	0.2002
29	93.609077	0.181315	94.10872	0.173414	93.59	0.1851	94.34	0.1674	92.44	0.2129	93.78	0.1846	64.57	0.6276	64.63	0.6273	71	0.5626	57.01	0.6838	96.53	0.1027	93.79	0.203
30	93.616901	0.182274	93.972587	0.176038	93.51	0.1856	94.23	0.1708	92.5	0.2124	93.93	0.1791	64.57	0.6278	64.45	0.629	71.03	0.5628	57.97	0.678	96.47	0.103	93.81	0.2024
31	93.674796	0.180946	93.654943	0.184009	93.71	0.1823	94.33	0.1675	92.56	0.2101	93.89	0.1806	64.57	0.628	64.3	0.629	71.18	0.5621	57.68	0.6838	96.57	0.1005	93.66	0.2086
32	93.612206	0.181486	93.8255	0.179219	93.74	0.1816	94.26	0.1706	92.61	0.2096	94.03	0.176	64.63	0.6274	64.62	0.6272	71.06	0.562	61.65	0.6529	96.61	0.0989	93.68	0.208
33	93.542836	0.182517	94.060213	0.174102	93.74	0.1816	94.26	0.1706	92.7	0.2075	93.73	0.183	64.58	0.6274	64.67	0.6275	71.2	0.5624	60.53	0.6605	96.64	0.098	93.7	0.2127
34	93.562656	0.182807	93.827065	0.184523	93.74	0.1816	94.26	0.1706	92.68	0.2075	93.7	0.1886	64.73	0.6269	64.49	0.6277	71.22	0.561	60.45	0.6628	96.64	0.098	93.7	0.2127
35	93.54388	0.181766	93.977281	0.173661	93.74	0.1816	94.26	0.1706	92.69	0.2069	93.7	0.1853	64.62	0.6273	64.7	0.6262	71.14	0.5617	58.84	0.6688	96.64	0.098	93.7	0.2127
36	93.61012	0.18271	94.133756	0.175037	93.74	0.1816	94.26	0.1706	92.74	0.2059	93.97	0.1778	64.63	0.6274	64.51	0.6291	71.19	0.5612	60.9	0.656	96.64	0.098	93.7	0.2127
37	93.521973	0.183827	94.060213	0.173673	93.74	0.1816	94.26	0.1706	92.79	0.2045	94.11	0.1757	64.67	0.6271	64.57	0.6262	71.23	0.5599	62.15	0.6485	96.64	0.098	93.7	0.2127
38	93.472423	0.184627	93.905303	0.177524	93.74	0.1816	94.26	0.1706	92.81	0.2042	93.93	0.1804	64.61	0.6272	64.19	0.629	71.16	0.5612	55.78	0.7147	96.64	0.098	93.7	0.2127
39	93.500066	0.183566	93.945986	0.17756	93.74	0.1816	94.26	0.1706	92.88	0.203	94.11	0.1772	64.71	0.6274	64.27	0.6292	71.13	0.5612	62.89	0.6453	96.64	0.098	93.7	0.2127
40	93.475031	0.184559	93.935033	0.177703	93.74	0.1816	94.26	0.1706	92.95	0.2013	94.16	0.1733	64.6	0.6272	64.58	0.6262	71.2	0.5605	55.72	0.6943	96.64	0.098	93.7	0.2127

Table A.5: Accuracy and Loss for Seed 5

Epoch	trainaccdarts	trainlossdarts	validaccdarts	validlossdarts	trainacclstm	trainlosslstm	validacclstm	validlosslstm	trainacclstmembed	trainlosslstmembed	validacclstmembed	validlosslstmembed	trainaccenn	trainlosscnn	validaccenn	validlosscnn	trainaccennembed	trainlosscnnembed	validaccennembed	validlosscnnembed	trainacchybrid	trainlosshybrid	validacchybrid	validlosshybrid
1	77.603861	0.466815	86.275585	0.359915	63.96	0.6282	79.65	0.4678	63.36	0.6254	81.58	0.4275	61.89	0.6481	63.49	0.6367	64.37	0.6245	60.61	0.6731	89.46	0.2597	92.92	0.2058
2	89.003522	0.289359	89.766541	0.276169	80.25	0.4469	86.27	0.347	83.91	0.3863	87.55	0.3217	63.2	0.6385	64.01	0.6321	67.97	0.595	54.83	0.8652	93.3	0.1928	93.61	0.1852
3	91.290131	0.239891	91.785068	0.226916	86.59	0.3377	90.14	0.2635	86.21	0.344	88.77	0.2952	63.45	0.6366	63.54	0.6361	68.85	0.5876	59.68	0.6837	93.76	0.1818	93.64	0.1827
4	92.124659	0.220084	91.633287	0.228747	89.64	0.2771	92.18	0.2227	87.44	0.3202	89.68	0.2746	63.67	0.6354	64.18	0.6333	69.21	0.5835	62.49	0.6479	93.91	0.1758	94.14	0.1717
5	92.640502	0.207472	92.872569	0.203627	91.18	0.2437	92.66	0.2128	88.57	0.2981	91.52	0.2352	63.71	0.6344	63.96	0.6327	69.59	0.5795	56.98	0.7279	94.2	0.1694	94.02	0.1754
6	92.948235	0.200061	93.04782	0.19798	91.84	0.2288	93.05	0.2026	89.63	0.275	91.97	0.2236	63.92	0.6342	63.66	0.6335	69.72	0.5775	54.12	0.8054	94.37	0.1648	94.18	0.1708
7	93.183467	0.195192	93.077551	0.199581	92.21	0.2208	93.4	0.1956	90.23	0.2627	92.34	0.2163	63.92	0.6335	64.15	0.6289	70.02	0.5751	57.01	0.7332	94.53	0.1595	94.38	0.1686
8	93.318557	0.191683	86.688679	0.393157	92.4	0.2155	93.61	0.1888	90.54	0.2548	92.36	0.2173	63.94	0.6332	64.01	0.632	70.17	0.5729	55.2	0.7729	94.65	0.156	94.24	0.1676
9	93.422351	0.188941	92.565878	0.218668	92.61	0.211	93.53	0.1942	90.84	0.2493	92.68	0.2107	63.98	0.6331	63.8	0.6361	70.38	0.5706	55.3	0.7477	94.83	0.1518	94.38	0.1649
10	93.529797	0.186705	93.503162	0.187787	92.74	0.208	93.81	0.1843	91.13	0.2439	92.86	0.2055	63.91	0.6329	64.34	0.6288	70.52	0.5699	52.3	0.782	94.96	0.1478	94.31	0.1685
11	93.565264	0.185808	93.689368	0.185564	92.83	0.2054	93.63	0.1847	91.24	0.2402	92.87	0.206	64.08	0.6324	64.26	0.6302	70.5	0.5698	56.74	0.7125	95.07	0.1443	94.22	0.1708
12	93.495372	0.185646	93.407713	0.190304	92.95	0.2039	93.92	0.1808	91.36	0.2375	93.09	0.2003	64.05	0.6322	64.36	0.6319	70.66	0.5685	53.48	0.7336	95.2	0.1407	94.35	0.1676
13	93.560048	0.184798	93.54854	0.185889	92.99	0.2025	93.42	0.192	91.54	0.2347	92.93	0.2017	64.12	0.6316	64.23	0.629	70.64	0.5682	59.3	0.6744	95.29	0.1376	94.14	0.1704
14	93.61012	0.183739	93.612695	0.188232	93.09	0.2	93.77	0.1824	91.62	0.2323	93.27	0.1956	64.14	0.6311	64.49	0.628	70.84	0.5671	53.89	0.793	95.44	0.1343	94.25	0.1723
15	93.616901	0.183725	93.243414	0.194935	93.09	0.1999	93.74	0.1856	91.72	0.2301	93.17	0.1961	64.15	0.6306	64.53	0.6289	70.82	0.566	51.65	0.8094	95.53	0.1311	93.9	0.1767
16	93.641415	0.18257	93.941292	0.179575	93.09	0.2008	93.76	0.1842	91.79	0.2276	92.85	0.2043	64.23	0.6307	64.53	0.6274	70.94	0.565	54.29	0.7365	95.59	0.1288	94.36	0.1721
17	93.630983	0.182883	93.387371	0.195395	93.18	0.1974	93.9	0.1799	91.81	0.2272	93.47	0.1895	64.31	0.63	64.35	0.6314	70.89	0.5647	51.84	0.8071	95.71	0.1255	94.16	0.1724
18	93.601775	0.182529	93.82863	0.179616	93.18	0.197	93.01	0.1971	91.91	0.2253	92.85	0.2061	64.31	0.6295	63.95	0.6308	70.96	0.564	54.33	0.7284	95.81	0.1233	94.22	0.1796
19	93.637764	0.182416	93.437443	0.190741	93.15	0.1989	93.97	0.1775	91.94	0.2239	93.48	0.1903	64.33	0.6294	64.31	0.6277	70.83	0.5647	53.4	0.7607	95.88	0.121	94.19	0.1762
20	93.584041	0.183627	93.98354	0.178788	93.26	0.1957	93.99	0.1772	91.97	0.223	93.43	0.1891	64.31	0.6291	64.65	0.6262	71.02	0.5629	57.66	0.6749	95.99	0.1183	94.12	0.1801
21	93.603861	0.181909	93.772299	0.180848	93.32	0.196	93.93	0.185	92.09	0.2212	92.69	0.2062	64.36	0.6287	64.39	0.6269	71.01	0.5635	53.85	0.7084	96.03	0.117	94.09	0.1815
22	93.587692	0.182278	93.960069	0.177727	93.32	0.1943	93.85	0.1784	92.12	0.2202	93.65	0.1853	64.56	0.6283	64.29	0.63	71.19	0.5618	51.66	0.784	96.13	0.1137	93.9	0.1864
23	93.618465	0.18226	93.808288	0.181445	93.31	0.1947	93.4	0.1903	92.22	0.2185	93.38	0.1908	64.48	0.6277	64.92	0.6255	71.12	0.5627	52.38	0.7706	96.17	0.1126	93.97	0.1853
24	93.611163	0.181752	93.944421	0.177375	93.4	0.1923	93.56	0.1941	92.21	0.2177	93.01	0.1961	64.63	0.6276	65.1	0.6244	71.12	0.5618	54.11	0.701	96.21	0.1108	93.96	0.1943
25	93.644023	0.182363	93.551669	0.186557	93.24	0.1964	94	0.1774	92.25	0.2172	93.48	0.1887	64.65	0.6272	65.13	0.6251	71.23	0.5618	52.74	0.7675	96.16	0.1116	94.01	0.182
26	93.564221	0.183461	93.916256	0.176606	93.39	0.1929	93.89	0.1778	92.33	0.2153	93.58	0.1926	64.68	0.6273	64.73	0.6247	71.07	0.5627	52.6	0.7305	96.36	0.1076	94.03	0.1885
27	93.599688	0.182299	93.981976	0.177527	93.42	0.1922	93.81	0.186	92.39	0.2147	93.49	0.1905	64.62	0.6272	64.73	0.6273	71.17	0.5611	53.96	0.7197	96.37	0.1068	93.99	0.1892
28	93.534491	0.183149	93.98354	0.176541	93.41	0.1919	93.93	0.1782	92.43	0.2136	93.78	0.1813	64.64	0.6264	64.97	0.626	71.21	0.5608	52.72	0.7538	96.41	0.1055	93.92	0.1921
29	93.553268	0.183357	93.572011	0.187203	93.35	0.1931	94.19	0.1743	92.32	0.2141	93.7	0.1859	64.7	0.6266	64.66	0.6251	71.26	0.5609	54.75	0.7177	96.43	0.1043	94.08	0.1964
30	93.538142	0.183715	93.922515	0.176328	93.44	0.1903	94.02	0.1808	92.41	0.213	93.63	0.1846	64.63	0.6267	64.8	0.6264	71.32	0.5608	52.68	0.7456	96.42	0.1047	93.85	0.197
31	93.511541	0.183727	93.778558	0.181222	93.43	0.1904	94.01	0.1788	92.41	0.2124	93.85	0.1801	64.81	0.6263	65	0.6239	71.44	0.5595	54.54	0.739	96.46	0.1026	93.95	0.1939
32	93.490678	0.183704	93.927209	0.177342	93.48	0.1902	93.89	0.1835	92.54	0.2105	93.93	0.1807	64.73	0.6264	65.01	0.6246	71.33	0.5596	54.36	0.7079	96.43	0.1042	93.92	0.1975
33	93.525103	0.184354	93.287227	0.197709	93.45	0.1908	93.79	0.1765	92.54	0.2104	93.63	0.1837	64.67	0.6265	64.06	0.6306	71.41	0.5595	53.74	0.7204	96.47	0.1027	93.9	0.2003
34	93.548574	0.183131	93.636166	0.187367	93.49	0.1893	93.84	0.1872	92.58	0.2099	93.35	0.1925	64.78	0.6262	64.86	0.6272	71.4	0.5593	52.46	0.7191	96.57	0.0999	93.94	0.2018
35	93.466685	0.185296	93.969458	0.175849	93.57	0.1892	94.12	0.1751	92.59	0.2095	93.72	0.1811	64.78	0.6264	64.81	0.6242	71.33	0.5591	53.84	0.7508	96.6	0.0995	93.7	0.2035
36	93.53084	0.184032	93.182389	0.198808	93.42	0.191	94.17	0.1756	92.68	0.2076	94.04	0.1773	64.81	0.6257	64.95	0.6254	71.49	0.5585	51.81	0.7438	96.53	0.1007	94.03	0.1998
37	93.444258	0.185716	94.088379	0.173957	93.34	0.1936	94.12	0.1744	92.72	0.2068	93.79	0.182	64.81	0.6263	64.7	0.6257	71.36	0.5588	50.57	0.8119	96.65	0.0976	93.79	0.2039
38	93.452603	0.185134	93.86149	0.177231	93.52	0.1885	94.08	0.1753	92.68	0.2058	93.95	0.1789	64.82	0.6258	64.91	0.6267	71.36	0.559	50.82	0.7663	96.71	0.0964	93.88	0.2071
39	93.444779	0.184969	93.207425	0.200336	93.37	0.1916	94.11	0.1751	92.75	0.2056	94.04	0.1759	64.81	0.6256	64.88	0.6247	71.49	0.5584	50.38	0.863	96.67	0.0974	93.65	0.2102
40	93.444258	0.185705	93.889655	0.178553	93.54	0.1891	94.08	0.177	92.71	0.2056	93.84	0.1805	64.75	0.626	65.15	0.6248	71.47	0.5581	51.71	0.804	96.61	0.0982	93.85	0.2028

Table A.6: Accuracy and Loss for Seed 6

Epoch	trainaccdarts	trainlossdarts	validaccdarts	validlossdarts	trainacclstm	trainlosslstm	validacclstm	validlosslstm	trainacclstmembed	trainlosslstmembed	validacclstmembed	validlosslstmembed	trainaccenn	trainlosscnn	validaccenn	validlosscnn	trainaccennembed	trainlosscnnembed	validaccennembed	validlosscnnembed	trainacchybrid	trainlosshybrid	validacchybrid	validlosshybrid
1	76.080325	0.486932	85.108282	0.357791	70.77	0.5564	82.22	0.3993	59.14	0.6653	66.3	0.6215	61.64	0.6495	63.35	0.6367	64.83	0.6223	63.58	0.6439	89.71	0.2583	93.34	0.1943
2	88.383363	0.300941	89.624149	0.283811	78.61	0.4649	83.46	0.3901	71.22	0.5624	81.75	0.4385	63.06	0.6395	62.48	0.6429	68.14	0.5934	63.23	0.643	93.34	0.1909	94.04	0.1789
3	91.135743	0.243083	92.558054	0.216182	83.32	0.3946	87.94	0.3087	80.98	0.437	85.65	0.3553	63.25	0.6381	63.37	0.6365	69	0.5853	62.49	0.6472	93.82	0.1802	93.96	0.1774
4	92.317644	0.216072	92.731741	0.203476	86.47	0.331	76.17	0.4945	84.28	0.3793	87.73	0.318	63.37	0.6377	63.21	0.6376	69.56	0.5793	61.68	0.6521	94.12	0.1727	94.13	0.1753
5	92.965447	0.202601	93.036867	0.197751	81.42	0.4213	89.87	0.2739	86.21	0.3434	88.96	0.293	63.3	0.6374	63.2	0.6374	69.82	0.5756	61.95	0.6484	94.3	0.1667	94.36	0.1686
6	93.186597	0.196242	92.792766	0.205328	88.59	0.2972	91.45	0.2369	87.65	0.3152	90.53	0.2582	63.34	0.637	63.81	0.6337	70.05	0.5738	60.16	0.6746	94.42	0.1622	94.37	0.1666
7	93.373323	0.192359	93.722227	0.184571	89.95	0.2711	90.92	0.2461	89.02	0.2882	91.59	0.2369	63.42	0.6368	63.63	0.6348	70.34	0.5707	61.37	0.6526	94.61	0.1582	94.12	0.1725
8	93.420265	0.190044	93.367029	0.198057	90.19	0.2644	91.97	0.2263	89.68	0.2737	91.95	0.2257	63.4	0.6367	63.71	0.6335	70.49	0.5697	59.41	0.6721	94.72	0.1541	94.28	0.1684
9	93.505283	0.187303	92.894475	0.2016	91.05	0.2484	91.95	0.2254	90.19	0.2631	91.88	0.2257	63.44	0.6364	63.47	0.6338	70.51	0.5684	58.24	0.6906	94.89	0.1496	94.29	0.1676
10	93.628897	0.185685	93.872443	0.178833	90.66	0.2537	92.63	0.2109	90.57	0.2551	92.01	0.2222	63.5	0.6364	63.56	0.6346	70.63	0.5677	62.54	0.6429	94.97	0.147	94.07	0.172
11	93.611163	0.184585	93.725357	0.184674	91.52	0.2357	92.62	0.2109	90.88	0.2495	91.97	0.2221	63.5	0.6362	62.79	0.6381	70.73	0.5665	61.58	0.6492	95.09	0.1426	94.28	0.1677
12	93.685228	0.183091	94.025789	0.176227	91.39	0.239	92.76	0.2062	91.04	0.2449	92.74	0.2075	63.58	0.6359	63.52	0.6349	70.87	0.5649	62.97	0.6426	95.22	0.1395	94.08	0.1732
13	93.733735	0.182354	93.606435	0.18449	91.56	0.2347	92.89	0.2053	91.3	0.2412	93.02	0.1996	63.5	0.6361	63.48	0.6362	70.87	0.5644	62.15	0.6469	95.35	0.1359	94.14	0.1763
14	93.730084	0.181837	93.830195	0.183727	92	0.2239	93.41	0.1927	91.33	0.2377	92.54	0.2084	63.56	0.6357	63.59	0.6348	70.94	0.5629	61.07	0.6553	95.49	0.1325	94.07	0.1766
15	93.74208	0.181934	94.028918	0.17429	92.32	0.2178	93.3	0.2005	91.53	0.2347	93.16	0.2049	63.61	0.6357	63.19	0.6362	70.96	0.5633	62.43	0.6467	95.64	0.1289	94.1	0.1738
16	93.714958	0.180741	93.994494	0.183243	92.45	0.2142	93.22	0.1966	91.71	0.2303	93.55	0.1911	63.54	0.6356	63.75	0.634	71	0.5628	61.79	0.6488	95.66	0.1258	94.08	0.1845
17	93.758771	0.180505	92.482946	0.223469	92.22	0.219	93.4	0.1958	91.78	0.2283	93.33	0.1941	63.5	0.6356	63.04	0.6384	71.19	0.5618	62.82	0.6431	95.73	0.1235	93.95	0.1836
18	93.74521	0.180122	93.930339	0.179599	92.5	0.2131	93.62	0.1862	91.85	0.2273	93.26	0.1951	63.62	0.6354	63.54	0.6353	71.01	0.5622	61.01	0.6543	95.9	0.1207	94.07	0.1891
19	93.77181	0.180118	94.04926	0.17891	92.79	0.207	93.63	0.1876	91.93	0.2246	93.56	0.1861	63.57	0.6357	62.91	0.6403	71.11	0.5617	60.38	0.6576	95.87	0.1206	94.03	0.1835
20	93.763986	0.179158	94.021095	0.174777	92.76	0.2061	91.77	0.2311	91.99	0.2227	93.75	0.1834	63.56	0.6352	63.48	0.6362	71.3	0.5607	60.77	0.654	95.96	0.1178	94.06	0.1832
21	93.746253	0.179424	94.293361	0.172485	92.7	0.208	91.77	0.2311	92.1	0.2205	93.82	0.1819	63.61	0.6352	63.84	0.6331	71.29	0.5607	62.72	0.6468	96.09	0.1144	94.04	0.187
22	93.753555	0.179865	93.404583	0.187715	92.7	0.208	91.77	0.2311	92.1	0.2196	93.81	0.1811	63.71	0.6349	63.69	0.6341	71.22	0.5603	63.38	0.6378	96.13	0.113	93.77	0.192
23	93.717566	0.17981	94.032048	0.172911	92.7	0.208	91.77	0.2311	92.24	0.2184	93.59	0.1858	63.62	0.635	63.52	0.6349	71.28	0.56	59.63	0.6619	96.19	0.1109	94.06	0.1978
24	93.759292	0.179296	94.086814	0.173524	92.7	0.208	91.77	0.2311	92.29	0.2169	93.93	0.1799	63.63	0.6348	63.75	0.6332	71.2	0.5597	57.63	0.6854	96.29	0.1091	93.95	0.1935
25	93.715479	0.180159	94.271455	0.169987	92.7	0.208	91.77	0.2311	92.23	0.2163	93.82	0.1799	63.63	0.6346	63.82	0.6327	71.38	0.5594	53.98	0.7075	96.29	0.109	93.69	0.1937
26	93.649239	0.181178	93.500033	0.184325	92.7	0.208	91.77	0.2311	92.38	0.2138	93.84	0.18	63.65	0.6347	63.81	0.6341	71.45	0.559	57.8	0.6748	96.39	0.1058	93.87	0.1969
27	93.668537	0.180344	93.780123	0.179378	92.7	0.208	91.77	0.2311	92.44	0.2134	93.61	0.1854	63.6	0.6343	63.59	0.6347	71.32	0.5592	63.63	0.6492	96.42	0.1048	94.07	0.1986
28	93.630462	0.18167	94.110285	0.171602	92.7	0.208	91.77	0.2311	92.43	0.212	93.88	0.1788	63.7	0.6341	63.8	0.633	71.39	0.5588	52.63	0.7131	96.42	0.1053	93.89	0.1942
29	93.707134	0.180482	94.169746	0.173767	92.7	0.208	91.77	0.2311	92.46	0.2125	93.89	0.1782	63.7	0.6342	63.65	0.6333	71.35	0.5583	54.87	0.6953	96.49	0.1026	93.85	0.2032
30	93.676882	0.181045	93.385806	0.188873	92.7	0.208	91.77	0.2311	92.54	0.2109	93.76	0.1821	63.74	0.6343	63.84	0.633	71.42	0.5581	57.03	0.6818	96.54	0.1016	93.57	0.2038
31	93.641415	0.181794	93.747263	0.179305	92.7	0.208	91.77	0.2311	92.54	0.2102	93.25	0.1941	63.69	0.6338	63.76	0.6336	71.41	0.5581	60.07	0.6603	96.55	0.1003	93.75	0.2201
32	93.681055	0.181071	93.241849	0.193981	92.7	0.208	91.77	0.2311	92.55	0.2097	93.72	0.1811	63.75	0.634	63.75	0.6333	71.47	0.5583	54.66	0.711	96.54	0.101	93.91	0.2032
33	93.596559	0.182705	94.269889	0.168701	92.7	0.208	91.77	0.2311	92.6	0.2085	93.82	0.1783	63.69	0.634	63.6	0.6347	71.44	0.5578	56.05	0.6982	96.44	0.1035	93.83	0.2034
34	93.666451	0.181389	94.257371	0.172378	92.68	0.2094	94.01	0.1788	92.61	0.2085	93.89	0.1795	63.65	0.634	63.75	0.6326	71.47	0.5574	54.31	0.7182	96.6	0.0985	93.8	0.1998
35	93.626811	0.182487	94.252677	0.171815	93.14	0.1971	94.16	0.175	92.62	0.208	93.96	0.1769	63.68	0.634	63.5	0.6333	71.57	0.5574	55.6	0.6937	96.63	0.0989	93.79	0.2059
36	93.558484	0.183063	94.202605	0.169825	93.35	0.193	93.82	0.1856	92.63	0.2076	93.98	0.1765	63.67	0.6341	63.53	0.6339	71.59	0.5573	55.52	0.6918	96.5	0.1017	93.56	0.2127
37	93.576217	0.183204	94.014835	0.174705	93.41	0.1911	94.26	0.1711	92.7	0.206	93.58	0.182	63.71	0.6339	63.34	0.6371	71.61	0.5569	58.3	0.6734	96.64	0.098	93.92	0.2017
38	93.589779	0.183294	93.958504	0.176381	93.53	0.1883	94.21	0.1709	92.71	0.2064	94.16	0.1733	63.62	0.6339	63.11	0.6364	71.65	0.5571	52.6	0.7186	96.63	0.0977	93.72	0.2071
39	93.563178	0.18391	94.241724	0.173025	93.44	0.1888	94.29	0.1702	92.78	0.2043	93.88	0.1771	63.62	0.6336	63.77	0.6335	71.51	0.5566	55.48	0.688	96.76	0.0949	93.68	0.2138
40	93.517279	0.184036	94.188523	0.170567	93.58	0.1866	94.18	0.1748	92.74	0.2046	93.89	0.1801	63.71	0.6336	64.04	0.6318	71.45	0.5573	54.57	0.6938	96.69	0.0962	93.83	0.2134

Table A.7: Accuracy and Loss for Seed 7

Epoch	trainaccdarts	trainlossdarts	validaccdarts	validlossdarts	trainacclstm	trainlosslstm	validacclstm	validlosslstm	trainacclstmembed	trainlosslstmembed	validacclstmembed	validlosslstmembed	trainaccenn	trainlosscnn	validaccenn	validlosscnn	trainaccennembed	trainlosscnnembed	validacccnnembed	validlosscnnembed	trainacchybrid	trainlosshybrid	validacchybrid	validlosshybrid
1	79.857088	0.432998	82.459162	0.396583	61.84	0.6446	55.67	0.6781	61.09	0.6541	60.2	0.6676	61.64	0.6502	62.89	0.6395	64.37	0.6255	63.45	0.6438	88.97	0.2677	92.65	0.211
2	89.125572	0.288577	91.032423	0.249139	68.56	0.5843	81.92	0.4286	67.14	0.6013	72.84	0.5609	62.83	0.6422	63.37	0.6375	67.85	0.5959	62.91	0.6401	93.25	0.1929	93.4	0.1912
3	91.513367	0.234229	92.567442	0.211397	84.29	0.3767	90.07	0.2658	73.94	0.5388	76.74	0.5094	62.98	0.6403	63.03	0.6396	68.52	0.5899	63.28	0.6449	93.75	0.1808	93.93	0.1775
4	92.759944	0.205604	91.298431	0.240893	89.4	0.2811	90.62	0.2551	77.25	0.4979	82.1	0.4192	63.03	0.6397	63.31	0.6367	69.23	0.5826	61.49	0.6555	94.11	0.1728	93.92	0.1789
5	93.233539	0.195141	93.213684	0.196601	91.17	0.2431	92.88	0.2033	83.46	0.3971	86.62	0.3354	63.06	0.6397	63.06	0.6387	69.5	0.5802	62.47	0.6497	94.26	0.1663	94.21	0.1716
6	93.367064	0.191085	93.251238	0.191462	91.8	0.2298	93.15	0.196	86.03	0.3481	88.14	0.3058	63.11	0.6389	63.54	0.6377	69.64	0.5782	59.99	0.6748	94.5	0.1606	94.23	0.1697
7	93.444779	0.18853	92.565877	0.212354	92.24	0.2194	92.33	0.2197	87.71	0.3155	90.38	0.2588	63.2	0.6387	63.19	0.6388	69.96	0.5751	57.4	0.7493	94.66	0.1563	94.52	0.1621
8	93.577261	0.185977	93.902173	0.17925	92.52	0.2141	93.66	0.1882	89.13	0.2847	91.42	0.2382	63.34	0.637	63.63	0.6326	70.1	0.5731	59.94	0.6644	94.82	0.1526	94.4	0.1645
9	93.64298	0.184177	93.521939	0.186481	92.54	0.2121	93.67	0.187	90.03	0.2671	91.7	0.2311	63.36	0.6366	63.33	0.6366	70.26	0.5713	58.39	0.6766	94.97	0.148	94.26	0.17
10	93.592386	0.183532	93.180824	0.191828	92.69	0.2087	93.42	0.1918	90.49	0.2583	91.86	0.2273	63.36	0.6363	63.26	0.6384	70.29	0.5715	55.65	0.7291	95.06	0.1447	94.47	0.1648
11	93.685228	0.182198	93.439008	0.194949	92.8	0.2052	93.62	0.1908	90.82	0.2503	92.38	0.2155	63.36	0.6361	63.71	0.6329	70.46	0.5697	62.82	0.6436	95.21	0.1411	94.41	0.1652
12	93.662278	0.182396	93.880267	0.178551	92.96	0.2025	93.72	0.1842	91.1	0.2446	92.38	0.2151	63.44	0.6364	63.69	0.6339	70.51	0.5694	56.09	0.7029	95.31	0.1372	94.34	0.1687
13	93.666451	0.181824	93.886526	0.178167	92.96	0.2023	93.86	0.182	91.33	0.239	92.4	0.2135	63.44	0.6358	63.67	0.6316	70.76	0.5675	60.41	0.6571	95.44	0.1336	94.33	0.1684
14	93.706613	0.181363	93.528199	0.182657	93.13	0.1991	93.14	0.202	91.47	0.2354	92.96	0.2021	63.49	0.6357	63.77	0.6335	70.69	0.567	54.41	0.7125	95.56	0.1305	93.7	0.1859
15	93.718087	0.181132	94.035177	0.176198	93.12	0.1983	93.81	0.1824	91.64	0.2322	93.11	0.1988	63.42	0.6354	63.77	0.6323	70.86	0.5661	61.55	0.653	95.63	0.1276	93.73	0.1828
16	93.657584	0.181294	93.942857	0.175756	93.16	0.1969	94.02	0.1762	91.78	0.2292	93.11	0.1998	63.61	0.6354	63.67	0.634	70.77	0.5664	61.95	0.6454	95.79	0.1239	94.19	0.1735
17	93.660713	0.181239	93.944422	0.176104	93.26	0.195	93.93	0.1797	91.96	0.2263	93.29	0.1955	63.53	0.6351	63.54	0.6339	70.83	0.5651	52.2	0.7684	95.87	0.1212	94.22	0.1807
18	93.707134	0.181076	93.636166	0.181253	93.31	0.1944	94.06	0.1752	91.97	0.2256	93.24	0.1951	63.63	0.635	63.64	0.6351	70.9	0.5652	53.04	0.7219	95.97	0.1179	94.32	0.1741
19	93.664886	0.181536	94.150969	0.171554	93.37	0.1923	93.93	0.1796	92.1	0.2219	93.22	0.1937	63.54	0.6351	63.61	0.6354	70.92	0.5648	53.94	0.7143	96.02	0.1159	93.96	0.1829
20	93.748339	0.180346	93.302874	0.188884	93.45	0.1905	94.11	0.1745	92.18	0.2201	93.5	0.1925	63.59	0.6348	63.53	0.6345	70.87	0.5649	58.21	0.6744	96.09	0.1137	94.06	0.1844
21	93.707656	0.180209	92.210679	0.215832	93.46	0.191	94.13	0.1773	92.2	0.2196	93.4	0.1949	63.56	0.6346	63.24	0.636	70.96	0.564	52.34	0.7709	96.2	0.1099	94.1	0.1835
22	93.681055	0.180321	93.334169	0.195041	93.46	0.1898	94.08	0.1742	92.3	0.2176	93.48	0.1917	63.65	0.6344	63.62	0.6323	71.05	0.563	57.13	0.6872	96.25	0.1092	94.14	0.1896
23	93.664364	0.181336	94.04926	0.173562	93.5	0.1887	94.16	0.1733	92.31	0.2171	93.22	0.198	63.77	0.6337	63.74	0.632	71.08	0.5636	52.97	0.7264	96.29	0.1075	93.91	0.1919
24	93.739993	0.180453	94.160357	0.171141	92.58	0.2073	94.16	0.1733	92.39	0.2147	93.68	0.1873	63.71	0.6329	63.7	0.631	71.19	0.5629	54.48	0.7028	96.38	0.1052	94.07	0.1946
25	93.667494	0.180036	93.844278	0.17851	92.58	0.2073	94.16	0.1733	92.43	0.2143	93.38	0.1949	63.77	0.6322	63.87	0.6326	71	0.5629	53.23	0.7034	96.46	0.1026	93.91	0.2035
26	93.628897	0.181744	93.781688	0.176755	92.58	0.2073	94.16	0.1733	92.45	0.2129	93.16	0.1946	63.77	0.6326	63.57	0.6321	71.08	0.5625	55.27	0.6892	96.24	0.1093	93.65	0.2017
27	93.650282	0.180985	93.705015	0.181529	92.58	0.2073	92.84	0.2079	92.51	0.2116	93.26	0.1979	63.77	0.6323	63.95	0.6296	71.08	0.5615	51.95	0.749	96.49	0.1014	93.97	0.2011
28	93.658105	0.18153	93.850537	0.176452	92.72	0.2068	93.96	0.1784	92.55	0.2113	93.4	0.1923	63.77	0.6323	63.89	0.631	71.03	0.5628	56.6	0.6725	96.41	0.1028	94.04	0.2007
29	93.612206	0.182099	94.147839	0.171847	93.17	0.1971	94.15	0.1745	92.52	0.2111	93.78	0.1835	63.78	0.6323	63.93	0.6296	71.09	0.5615	53.41	0.6996	96.57	0.099	93.92	0.2037
30	93.572566	0.182407	93.606436	0.190582	93.42	0.1914	94.15	0.1774	92.61	0.2093	93.46	0.1911	63.82	0.6323	63.66	0.6319	71.24	0.5618	52.65	0.6906	96.58	0.0984	93.75	0.2023
31	93.592387	0.181737	93.742569	0.178083	93.42	0.1914	94.15	0.1774	92.6	0.2092	93.76	0.1822	63.83	0.6321	63.88	0.629	71.19	0.5604	50.15	0.7619	96.67	0.096	93.97	0.2069
32	93.509455	0.1833	94.252677	0.169539	93.42	0.1914	94.15	0.1774	92.64	0.209	93.91	0.1812	63.86	0.6319	63.92	0.6289	71.3	0.5605	53.11	0.7115	96.6	0.0977	93.98	0.2081
33	93.574131	0.182047	94.161922	0.170434	93.42	0.1914	93.12	0.2039	92.76	0.2062	93.69	0.1864	63.78	0.6322	63.83	0.6302	71.3	0.5605	56.68	0.6771	96.58	0.0966	93.83	0.2082
34	93.549095	0.183366	94.238594	0.170864	92.68	0.2094	94.01	0.1788	92.75	0.2064	93.82	0.1827	63.82	0.6318	63.91	0.6302	71.32	0.5613	51.89	0.7172	96.58	0.0966	93.83	0.2082
35	93.508934	0.183756	94.208864	0.173306	93.14	0.1971	94.16	0.175	92.72	0.2056	93.92	0.1807	63.87	0.6314	63.94	0.6291	71.28	0.5605	51.19	0.6984	96.58	0.0966	93.83	0.2082
36	93.480768	0.184407	93.834889	0.176817	93.35	0.193	93.82	0.1856	92.85	0.2047	93.87	0.1809	63.99	0.6314	64.25	0.6285	71.25	0.5606	50.35	0.7181	96.58	0.0966	93.83	0.2082
37	93.516757	0.184486	94.199476	0.17042	93.41	0.1911	94.26	0.1711	92.75	0.2055	93.83	0.1812	63.97	0.6311	63.48	0.6327	71.38	0.5597	52.63	0.6963	96.58	0.0966	93.83	0.2082
38	93.48807	0.184232	94.183828	0.170086	93.53	0.1883	94.21	0.1709	92.82	0.2041	93.98	0.181	63.96	0.6307	64.21	0.6281	71.33	0.5601	50.5	0.741	96.58	0.0966	93.83	0.2082
39	93.516757	0.184194	94.251113	0.171066	93.44	0.1888	94.29	0.1702	92.81	0.2045	93.95	0.1788	64.06	0.6309	63.96	0.6313	71.34	0.5603	57.91	0.6728	96.58	0.0966	93.83	0.2082
40	93.479204	0.18491	92.606561	0.214524	93.58	0.1866	94.18	0.1748	92.87	0.2036	93.86	0.1796	64.14	0.6305	64.05	0.6286	71.29	0.5605	50.56	0.7353	96.58	0.0966	93.83	0.2082

Table A.8: Accuracy and Loss for Seed 8

Epoch	trainaccdarts	trainlossdarts	validaccdarts	validlossdarts	trainacclstm	trainlosslstm	validacclstm	validlosslstm	trainacclstmembed	trainlosslstmembed	validacclstmembed	validlosslstmembed	trainaccenn	trainlosscnn	validaccenn	validlosscnn	trainaccennembed	trainlosscnnembed	validaccennembed	validlosscnnembed	trainacchybrid	trainlosshybrid	validacchybrid	validlosshybrid
1	77.287262	0.468698	86.602618	0.350734	62.38	0.6415	63.3	0.636	63.13	0.6356	76.45	0.5156	60.51	0.6563	62.87	0.6436	64.64	0.6222	58.19	0.7337	86.73	0.3046	93.15	0.1979
2	88.671797	0.295135	90.667837	0.252849	81.61	0.4132	89.96	0.2711	74.54	0.5116	84.21	0.3869	62.17	0.6457	62.69	0.6418	68.33	0.5915	56.91	0.7343	93.28	0.1927	92.64	0.2098
3	90.843136	0.248033	90.558304	0.250005	89.81	0.2704	92.28	0.2157	84.16	0.3823	86.72	0.3368	62.26	0.645	62.99	0.6415	69.19	0.5829	55.1	0.7456	93.83	0.1801	94.03	0.1764
4	91.723564	0.229763	92.277963	0.215048	91.48	0.2339	91.95	0.225	85.89	0.3504	88.09	0.3103	62.25	0.6446	62.87	0.6409	69.63	0.5786	51.82	0.7774	94.07	0.1724	93.88	0.1791
5	92.233669	0.21628	92.89604	0.202019	92.12	0.2196	92.74	0.2087	87.06	0.3271	89.42	0.279	62.5	0.6436	62.63	0.6403	69.98	0.5753	54.25	0.7364	94.31	0.1654	94.27	0.1678
6	92.684837	0.205788	93.165176	0.198509	92.52	0.211	93.36	0.1916	88.69	0.2961	91.73	0.2354	62.44	0.6432	62.84	0.6391	69.99	0.5739	51.18	0.8131	94.49	0.16	93.78	0.1744
7	92.982137	0.198907	93.356076	0.191213	92.75	0.2052	93.1	0.1959	89.76	0.2732	92.25	0.2227	62.59	0.643	61.75	0.6445	70.23	0.5718	52.31	0.8086	94.66	0.1554	94.37	0.1662
8	93.124007	0.194097	93.306004	0.194116	92.93	0.2009	93.64	0.1862	90.32	0.2614	91.74	0.2286	62.56	0.643	62.96	0.6378	70.44	0.5702	51.08	0.8855	94.81	0.151	94.34	0.1653
9	93.303952	0.191306	92.963324	0.198313	93.05	0.1977	93.94	0.1798	90.43	0.257	92.49	0.2129	62.52	0.6422	62.9	0.6398	70.44	0.5686	51.37	0.8275	94.92	0.1477	94.44	0.1651
10	93.47138	0.189515	93.751957	0.183264	93.13	0.1955	93.98	0.1764	90.87	0.2485	92.54	0.2116	62.57	0.642	61.86	0.6443	70.63	0.5674	52.32	0.7973	95.09	0.143	94.27	0.1681
11	93.472423	0.186859	93.845842	0.179066	93.22	0.1944	93.96	0.1774	91.17	0.2424	92.93	0.2048	62.74	0.6407	63.06	0.6378	70.65	0.5671	51.54	0.8391	95.24	0.1393	94.42	0.164
12	93.4912	0.186382	93.70658	0.184754	93.3	0.1917	94.19	0.1724	91.17	0.2424	92.93	0.2048	62.8	0.64	62.83	0.6382	70.83	0.5652	51.98	0.8081	95.34	0.1359	94.39	0.1648
13	93.621073	0.184027	93.894349	0.178959	93.38	0.1897	94.06	0.1755	91.17	0.2424	92.93	0.2048	62.94	0.6399	62.92	0.6373	70.83	0.5651	51.14	0.8457	95.49	0.1327	94.31	0.1663
14	93.573609	0.183417	93.867749	0.178083	93.46	0.1884	94.06	0.1736	91.17	0.2424	91.21	0.2447	63	0.64	63.3	0.6368	70.78	0.5649	51.23	0.7657	95.64	0.1283	94.25	0.168
15	93.601253	0.182592	94.025788	0.17804	93.42	0.1879	94.05	0.1734	91.17	0.2424	90.45	0.2574	63.05	0.6394	63.28	0.6369	70.98	0.5639	50.7	0.8244	95.74	0.1247	94.09	0.1733
16	93.607512	0.183645	93.398324	0.193946	93.18	0.194	94.31	0.1703	91.17	0.2424	92.54	0.2141	62.98	0.6394	63.22	0.636	70.92	0.5638	52.62	0.7712	95.86	0.1212	94.15	0.1767
17	93.678969	0.182005	93.935033	0.177541	93.58	0.1857	94.05	0.1737	90.72	0.2516	93.04	0.2011	62.97	0.6389	62.49	0.6391	70.97	0.5633	53.04	0.786	95.99	0.1177	94.17	0.1789
18	93.659149	0.181747	92.983666	0.207564	93.57	0.1851	94.29	0.1681	91.29	0.2398	92.16	0.2242	62.93	0.6394	63.04	0.6367	71.02	0.5626	52.01	0.7686	96.08	0.1154	94.03	0.1881
19	93.683141	0.181188	94.10872	0.173099	93.59	0.1849	93.97	0.1781	91.51	0.2338	93.2	0.1972	63.08	0.6395	63.19	0.6358	71.09	0.562	51.77	0.7914	96.2	0.1123	94.23	0.1784
20	93.680012	0.180932	94.025789	0.172937	93.63	0.1837	94.13	0.1762	91.73	0.2292	93.28	0.1969	63.01	0.6389	63.02	0.6371	71.15	0.5621	53.67	0.72	96.3	0.1092	93.87	0.1877
21	93.606469	0.183077	93.046255	0.206639	93.35	0.1894	94.01	0.1754	91.91	0.2255	93.27	0.1935	62.96	0.6391	62.96	0.6379	71.07	0.562	51.86	0.7993	96.38	0.1068	93.94	0.1971
22	93.666451	0.181488	94.085249	0.171157	93.5	0.1858	94.19	0.1713	91.84	0.2259	93.45	0.1905	63.01	0.6389	63.1	0.6379	71.18	0.5606	53.45	0.7447	96.42	0.105	94.06	0.1884
23	93.69253	0.181737	94.152533	0.171804	93.62	0.1831	93.97	0.1756	91.87	0.2261	93.6	0.1869	63.12	0.6388	62.96	0.6389	71.14	0.5614	51.12	0.8201	96.49	0.1031	93.9	0.1917
24	93.645066	0.181519	93.579835	0.18648	93.62	0.1829	94.39	0.1666	92.1	0.2217	93.35	0.1917	63.08	0.639	63.25	0.6355	71.29	0.5601	53.24	0.7854	96.6	0.0997	93.71	0.2077
25	93.620552	0.181308	94.086814	0.170836	93.68	0.1821	94.22	0.1704	92.19	0.2197	93.41	0.1944	63.13	0.6387	63.23	0.636	71.17	0.5606	51.52	0.7866	96.6	0.0996	93.84	0.198
26	93.693573	0.18125	94.085249	0.171601	93.71	0.1824	93.36	0.1902	92.22	0.2172	93.77	0.1843	63.1	0.6387	62.72	0.6393	71.26	0.5593	53.48	0.7684	96.71	0.0972	93.89	0.2029
27	93.636721	0.182073	94.119674	0.177214	93.72	0.1807	94.15	0.1714	92.3	0.2166	93.67	0.1847	63.09	0.6386	63.06	0.6366	71.27	0.5601	53.86	0.7412	96.74	0.0951	93.95	0.2016
28	93.694095	0.180939	93.751957	0.179908	93.7	0.1809	94.43	0.1653	92.29	0.2161	93.6	0.1865	63.18	0.6383	62.91	0.6397	71.34	0.5591	50.38	0.8269	96.82	0.0935	93.76	0.2032
29	93.684706	0.181302	94.105591	0.172901	93.76	0.1804	94.43	0.1641	92.35	0.2152	93.81	0.1817	63.19	0.6386	63.23	0.6354	71.29	0.5596	50.57	0.8366	96.87	0.0913	93.77	0.2159
30	93.636721	0.182049	94.044565	0.171991	93.8	0.1799	94.27	0.1664	92.49	0.2125	93.74	0.1808	63.15	0.6383	62.92	0.6388	71.4	0.5591	53.07	0.7348	96.91	0.0909	93.76	0.2051
31	93.618987	0.181941	93.543846	0.185457	93.78	0.1799	94.4	0.1672	92.41	0.2124	93.8	0.182	63.11	0.6383	63.21	0.6358	71.3	0.5594	52.3	0.7469	96.92	0.0901	93.61	0.2138
32	93.58352	0.182793	94.053954	0.177002	92.51	0.202	94.37	0.1667	92.57	0.2102	93.85	0.1793	63.16	0.6383	62.9	0.6388	71.31	0.559	51.54	0.7748	97.01	0.0877	93.61	0.2221
33	93.589257	0.181845	94.130627	0.170773	93.21	0.19	94.25	0.169	92.5	0.2104	93.92	0.1785	63.16	0.6382	62.98	0.6371	71.38	0.5589	52.31	0.7508	97.03	0.0865	93.73	0.2181
34	93.54075	0.184131	94.083685	0.170783	93.75	0.179	94.02	0.1759	92.67	0.2077	94.06	0.1745	63.16	0.6383	63.13	0.636	71.35	0.5588	51.53	0.835	97.03	0.0869	93.7	0.2109
35	93.606469	0.181815	94.20417	0.16834	93.83	0.178	94.33	0.1658	92.7	0.2073	93.97	0.1764	63.11	0.638	63	0.6371	71.34	0.5585	51.85	0.7635	97.09	0.0843	93.6	0.2269
36	93.555354	0.182901	94.140015	0.171587	93.86	0.1781	94.3	0.1713	92.7	0.2058	93.92	0.1806	63.21	0.638	63.11	0.6359	71.34	0.5584	52.26	0.7506	97	0.088	93.29	0.227
37	93.512585	0.184413	94.211994	0.167123	93.72	0.1805	94.37	0.1654	92.77	0.2053	93.55	0.1849	63.17	0.6377	63.08	0.6378	71.36	0.5587	50.15	0.913	97.07	0.0854	93.71	0.2158
38	93.525624	0.183614	94.119673	0.174041	93.9	0.1769	94.3	0.167	92.67	0.2069	94.02	0.1755	63.16	0.6383	63.18	0.6363	71.34	0.5586	53.23	0.7348	97.18	0.0823	93.91	0.2228
39	93.484419	0.184368	93.819242	0.176738	93.86	0.1776	94.43	0.1653	92.73	0.2044	94.07	0.1748	63.11	0.6382	63.06	0.6375	71.41	0.5587	51.3	0.8654	97.17	0.0829	93.57	0.2204
40	93.515714	0.184484	93.503162	0.186589	93.89	0.1774	94.34	0.167	92.62	0.2073	94.12	0.174	63.11	0.6381	63.2	0.6365	71.51	0.5578	52.15	0.7566	97.16	0.0822	93.77	0.2337

Table A.9: Accuracy and Loss for Seed 9

Epoch	trainaccdarts	trainlossdarts	validaccdarts	validlossdarts	trainacclstm	trainlosslstm	validacclstm	validlosslstm	trainacclstmembed	trainlosslstmembed	validacclstmembed	validlosslstmembed	trainaccenn	trainlosscnn	validaccenn	validlosscnn	trainaccennembed	trainlosscnnembed	validaccennembed	validlosscnnembed	trainacchybrid	trainlosshybrid	validacchybrid	validlosshybrid
1	79.229627	0.443306	87.391251	0.32724	67.57	0.5888	83.53	0.3948	69.25	0.5835	79.27	0.502	62.05	0.647	63.19	0.6371	64.61	0.6239	59.47	0.6986	89	0.2682	92.32	0.2229
2	88.697876	0.298189	90.33298	0.263447	85.18	0.3598	89.87	0.2691	82.29	0.42	85.36	0.3545	63.24	0.6387	63	0.6393	67.72	0.5971	57.18	0.7375	93.4	0.19	93.78	0.1832
3	90.796715	0.251103	91.179509	0.241918	89.38	0.2795	92.06	0.2247	84.82	0.3719	87.74	0.3154	63.44	0.6373	63.43	0.6365	68.79	0.588	59.08	0.7064	93.9	0.1781	94.04	0.1743
4	91.698006	0.229128	91.107531	0.242301	90.83	0.2499	91.81	0.2287	86.4	0.3426	89.31	0.2841	63.44	0.6372	63.88	0.6349	69.49	0.5808	56.27	0.7361	94.14	0.171	94.27	0.1699
5	92.274353	0.215855	92.842838	0.20314	91.67	0.2317	92.99	0.2027	87.81	0.3151	90.9	0.2494	63.53	0.6364	63.65	0.6338	69.77	0.5767	58.38	0.6957	94.33	0.1656	94.11	0.1729
6	92.795933	0.203728	91.744384	0.228584	92.04	0.2225	93.25	0.1931	88.96	0.2904	91	0.2479	63.46	0.6367	63.41	0.6387	70.1	0.5741	55.57	0.7287	94.48	0.1606	94.31	0.1699
7	93.094799	0.196957	93.3639	0.189392	92.32	0.2165	93.34	0.1968	89.61	0.2763	91.63	0.2311	63.56	0.6359	63.46	0.6344	70.29	0.5724	52.08	0.7873	94.65	0.1564	94.25	0.1687
8	93.271093	0.193419	93.443702	0.188563	92.44	0.2137	93.5	0.1915	90.02	0.2674	91.59	0.2378	63.61	0.6361	63.22	0.6367	70.33	0.5709	55.38	0.7448	94.83	0.152	94.34	0.171
9	93.317514	0.190784	93.701885	0.18234	92.64	0.2095	93.39	0.1924	90.44	0.26	92.28	0.2213	63.62	0.6358	63.73	0.6344	70.54	0.5698	52.9	0.7603	94.96	0.1477	94.33	0.1679
10	93.415049	0.189658	93.262191	0.193359	92.7	0.2072	93.71	0.1878	90.63	0.255	92.51	0.2179	63.65	0.6356	63.63	0.6362	70.61	0.5674	53.4	0.7297	95.11	0.1432	94.09	0.1705
11	93.436434	0.187653	92.989924	0.200032	92.91	0.2035	93.86	0.1811	90.86	0.2492	92.45	0.2134	63.64	0.6353	63.48	0.6354	70.77	0.5669	53.48	0.7431	95.25	0.1395	94.29	0.1704
12	93.487027	0.186814	93.841148	0.182084	92.97	0.2018	93.82	0.1815	91.04	0.2461	92.64	0.2109	63.79	0.6347	63.97	0.6335	70.84	0.5661	51.49	0.8143	95.4	0.1356	94.06	0.1746
13	93.519365	0.1855	92.173125	0.223379	93.04	0.2001	94	0.1787	91.18	0.2422	92.59	0.2146	63.81	0.6341	63	0.6409	70.84	0.5659	57.11	0.7236	95.47	0.1326	94.01	0.1797
14	93.500067	0.185409	93.562623	0.183145	93.08	0.1992	93.77	0.1846	91.29	0.2396	92.39	0.2212	63.74	0.6339	63.86	0.6342	70.89	0.5652	57.83	0.6813	95.61	0.1288	94	0.1777
15	93.502153	0.184149	93.811418	0.181465	93.15	0.1973	93.91	0.1789	91.37	0.237	93.14	0.1992	63.85	0.6337	63.7	0.6343	71	0.5643	53.24	0.8233	95.73	0.1252	93.95	0.185
16	93.542836	0.184468	93.406147	0.191241	93.17	0.1962	93.93	0.1779	91.52	0.2348	93.32	0.1959	63.83	0.634	64.06	0.6333	71.06	0.5633	55.5	0.7232	95.81	0.1223	93.82	0.1868
17	93.587692	0.183759	93.928774	0.176739	93.19	0.1964	93.85	0.1847	91.65	0.2317	93.4	0.1925	63.89	0.6334	62.99	0.6375	70.95	0.5638	56.95	0.6942	95.9	0.1204	94.17	0.1822
18	93.58039	0.183932	93.906868	0.178311	93.18	0.1965	93.97	0.178	91.8	0.2288	93.32	0.1966	63.74	0.6335	63.85	0.6334	71.07	0.5636	51.73	0.821	95.97	0.1181	94.07	0.1831
19	93.569959	0.183616	93.969458	0.17561	93.3	0.1936	94.13	0.1741	91.86	0.227	93.59	0.1895	63.82	0.6333	63.92	0.6328	71.11	0.5634	52.45	0.8647	96.06	0.1159	94.02	0.1866
20	93.589257	0.183198	93.892785	0.182432	93.17	0.1963	93.92	0.1779	91.98	0.2257	93.51	0.1898	63.84	0.6332	63.72	0.6322	71.07	0.5628	51.41	0.8843	96.15	0.1128	94.05	0.1815
21	93.554833	0.18363	94.043001	0.173656	93.32	0.1917	94.04	0.1786	92.02	0.2242	93.61	0.1894	63.8	0.6334	64.11	0.6312	71.08	0.5627	52.98	0.7374	96.18	0.1114	93.7	0.1944
22	93.523538	0.183489	92.750518	0.210519	93.34	0.1919	94.12	0.1767	92.06	0.2226	93.06	0.2066	64	0.6332	63.77	0.6343	71.1	0.5621	53	0.731	96.1	0.1132	93.93	0.1846
23	93.591343	0.183306	94.032048	0.175847	93.38	0.1901	93.86	0.1808	92.1	0.222	93.48	0.1889	63.86	0.6332	63.96	0.634	71.23	0.562	53.03	0.7263	96.15	0.1128	94.02	0.1905
24	93.488592	0.184362	94.005447	0.173318	93.41	0.1914	93.98	0.176	92.17	0.2185	93.68	0.1841	63.9	0.6332	64.09	0.6332	71.18	0.563	54.1	0.7342	96.22	0.1108	94.06	0.1961
25	93.503196	0.183768	94.0164	0.175689	93.42	0.1906	93.94	0.179	92.29	0.2169	93.68	0.1865	63.9	0.6334	64.07	0.6327	71.28	0.561	53.55	0.7826	96.35	0.1064	94.04	0.1912
26	93.500067	0.183796	93.770734	0.179341	93.49	0.1891	93.85	0.1807	92.21	0.2198	93.64	0.1848	63.9	0.6328	62.99	0.6416	71.26	0.5614	53.47	0.7701	96.46	0.1036	93.53	0.2062
27	93.472944	0.184522	93.853666	0.176812	93.5	0.1883	94.16	0.1727	92.38	0.2146	93.78	0.1824	63.88	0.633	63.94	0.6323	71.37	0.5596	52.08	0.7311	96.42	0.1053	94.05	0.1954
28	93.529275	0.184065	93.371723	0.187311	93.55	0.1874	94.16	0.1735	92.35	0.2161	93.72	0.1878	63.95	0.6326	63.59	0.6348	71.45	0.5594	52.33	0.7985	96.55	0.1008	93.69	0.2019
29	93.512585	0.184983	93.938163	0.177341	93.46	0.1892	94.3	0.1731	92.44	0.2129	93.78	0.1846	63.9	0.6329	63.65	0.6334	71.44	0.5593	52.24	0.7529	96.5	0.1026	93.92	0.2047
30	93.450516	0.185307	94.122803	0.171759	93.5	0.1881	94.16	0.1737	92.5	0.2124	93.93	0.1791	63.9	0.6329	63.83	0.634	71.41	0.5592	51.2	0.9064	96.62	0.0988	93.85	0.2015
31	93.428089	0.185929	93.772299	0.180778	93.53	0.1867	94.25	0.1713	92.56	0.2101	93.89	0.1806	63.95	0.6323	63.98	0.6315	71.45	0.5594	55.5	0.7199	96.57	0.1	93.85	0.2003
32	93.469815	0.185852	93.992929	0.173117	93.6	0.1866	94.31	0.1696	92.61	0.2096	94.03	0.176	64.01	0.6326	64.07	0.6338	71.52	0.5591	50.29	0.8299	96.59	0.0986	93.74	0.1998
33	93.497459	0.185305	93.823936	0.179986	93.61	0.1849	94.15	0.1725	92.7	0.2075	93.73	0.183	63.93	0.6325	64.09	0.6322	71.34	0.5599	50.43	0.8268	96.63	0.0977	93.72	0.2045
34	93.467207	0.185575	94.017965	0.176261	93.55	0.1863	94.16	0.1772	92.68	0.2075	93.7	0.1886	63.97	0.6323	64.09	0.632	71.33	0.5591	50.81	0.7827	96.64	0.0972	93.76	0.2084
35	93.465642	0.185657	93.653378	0.182188	93.65	0.1838	94.32	0.1711	92.69	0.2069	93.7	0.1853	63.94	0.6325	63.52	0.636	71.52	0.5589	52.49	0.7419	96.64	0.0972	93.76	0.2084
36	93.434348	0.185823	94.088378	0.17241	93.65	0.1837	93.87	0.1781	92.74	0.2059	93.97	0.1778	63.96	0.6326	63.49	0.6355	71.4	0.5592	51.41	0.7602	96.64	0.0972	93.76	0.2084
37	93.457819	0.186059	93.850536	0.17532	93.63	0.1845	94.25	0.1703	92.79	0.2045	94.11	0.1757	63.81	0.6327	64.24	0.6302	71.44	0.5585	52.05	0.8522	96.64	0.0972	93.76	0.2084
38	93.449473	0.187431	93.870878	0.179082	93.66	0.1838	94.31	0.1691	92.81	0.2042	93.93	0.1804	64.09	0.6309	64.14	0.6293	71.46	0.5583	51.31	0.7879	96.64	0.0972	93.04	0.1996
39	93.361326	0.187609	94.086814	0.172516	93.67	0.1836	94.31	0.1694	92.88	0.203	94.11	0.1772	64.15	0.6306	64.37	0.6287	71.65	0.5582	50.23	0.8727	96.64	0.0972	93.78	0.1839
40	93.395229	0.187452	94.086814	0.170581	93.72	0.1826	93.85	0.1788	92.95	0.2013	94.16	0.1733	64.06	0.6309	64.37	0.6288	71.45	0.5585	52.46	0.7743	96.64	0.0972	93.35	0.1922

Table A.10: Accuracy and Loss for Seed 10

Epoch	trainaccdarts	trainlossdarts	validaccdarts	validlossdarts	trainacclstm	trainlosslstm	validacclstm	validlosslstm	trainacclstmembed	trainlosslstmembed	validacclstmembed	validlosslstmembed	trainaccenn	trainlosscnn	validaccenn	validlosscnn	trainaccennembed	trainlosscnnembed	validaccennembed	validlosscnnembed	trainacchybrid	trainlosshybrid	validacchybrid	validlosshybrid
1	78.983963	0.445655	85.573012	0.368654	61.05	0.6554	64.18	0.6367	59.06	0.6592	59.68	0.6621	59.55	0.6634	60.31	0.657	64.03	0.6277	59.11	0.6688	89.46	0.2619	93.61	0.1875
2	88.987353	0.291626	91.032423	0.245106	75.52	0.5064	83.55	0.3963	67.44	0.592	84.68	0.38	61.74	0.6499	62.81	0.6434	67.79	0.5966	60.38	0.6942	93.49	0.1884	93.96	0.1787
3	91.217631	0.241718	91.758466	0.231211	81.19	0.4193	67.55	0.5996	84.05	0.3843	86.66	0.328	62.45	0.644	62.86	0.6414	68.6	0.589	63.08	0.6387	93.91	0.1773	94.14	0.1728
4	91.921765	0.224546	92.484511	0.209213	82.79	0.388	90.22	0.2603	86.07	0.3448	88.21	0.3086	62.68	0.642	62.81	0.64	69.01	0.5846	63.58	0.6394	94.15	0.1703	93.97	0.1803
5	92.448039	0.212674	92.83345	0.202568	90.1	0.2633	92.2	0.2232	87.5	0.3166	90.22	0.2606	62.91	0.64	63.44	0.6357	69.55	0.5797	61.66	0.6492	94.36	0.1647	94.18	0.1696
6	92.740124	0.206488	92.119924	0.217502	91.52	0.2338	92.84	0.2046	89.07	0.2879	91.85	0.2295	63.09	0.6392	63.22	0.6398	69.66	0.5781	60.51	0.6752	94.54	0.1601	94.13	0.1731
7	92.872084	0.201154	93.302874	0.191272	92.04	0.2217	93.37	0.1937	90.01	0.2672	92.29	0.2164	63.04	0.6389	63.48	0.6365	69.91	0.5754	61.08	0.6593	94.71	0.155	93.99	0.1754
8	92.999349	0.198758	92.245104	0.217804	92.5	0.2123	93.45	0.1905	90.52	0.2554	92.64	0.2124	62.99	0.6389	63.51	0.6351	70.06	0.5738	62.06	0.6462	94.87	0.151	94.26	0.1697
9	93.174079	0.19512	92.939852	0.20298	92.71	0.2073	93.6	0.1863	90.93	0.2469	93.03	0.2042	63.19	0.6383	63.49	0.6362	70.14	0.5737	62.1	0.6499	95.01	0.1471	94.31	0.1668
10	93.166255	0.194514	93.637731	0.185088	92.9	0.2024	93.69	0.1855	91.15	0.2413	92.93	0.2087	63.13	0.6383	63.45	0.6363	70.21	0.5726	62.63	0.6431	95.17	0.1428	94.14	0.1694
11	93.179295	0.193286	93.79577	0.183082	92.99	0.2001	93.63	0.1837	91.44	0.2358	93.08	0.1984	63.22	0.638	63.59	0.6348	70.21	0.5713	62.64	0.643	95.29	0.1389	93.91	0.1782
12	93.27005	0.191878	93.409277	0.190483	92.99	0.2001	93.63	0.1837	91.56	0.2344	93.37	0.1945	63.28	0.6375	63.32	0.6354	70.54	0.5694	60.54	0.6578	95.38	0.1353	94.18	0.1717
13	93.300301	0.191137	93.069727	0.197159	92.99	0.2001	93.63	0.1837	91.67	0.2302	93.44	0.1922	63.35	0.637	63.66	0.6337	70.54	0.5688	62.8	0.6439	95.55	0.1313	94.22	0.1708
14	93.258054	0.191268	93.809853	0.188743	92.99	0.2001	92.36	0.2126	91.79	0.2283	93.6	0.1892	63.36	0.637	63.44	0.6379	70.42	0.5688	63.52	0.6401	95.66	0.1279	94.16	0.1781
15	93.367585	0.189385	93.04782	0.19631	92.41	0.2134	93.49	0.1881	91.94	0.2252	93.55	0.1901	63.3	0.6366	63.74	0.635	70.45	0.5688	62.94	0.6439	95.74	0.1249	94.2	0.1734
16	93.337334	0.189849	93.285662	0.192345	93.04	0.1998	93.23	0.1936	92.07	0.2221	93.73	0.1851	63.4	0.6366	63.27	0.6347	70.68	0.567	62.76	0.6421	95.82	0.1222	93.86	0.1865
17	93.303952	0.189906	92.67228	0.204503	93.2	0.1954	93.86	0.1791	92.14	0.2209	93.59	0.1869	63.39	0.6364	63.69	0.6336	70.64	0.5665	62.3	0.6461	95.96	0.1185	94.26	0.1774
18	93.334726	0.188254	93.656508	0.184786	93.33	0.1931	93.99	0.1786	92.14	0.2207	93.54	0.1905	63.45	0.636	63.74	0.6348	70.75	0.5664	59.26	0.6662	96.03	0.1169	93.86	0.1906
19	93.263269	0.189704	93.290357	0.192046	93.38	0.1914	94.02	0.1797	92.22	0.2186	93.61	0.1862	63.41	0.636	63.61	0.6355	70.66	0.5671	63.08	0.6441	96.18	0.1136	94.1	0.1859
20	93.323773	0.189758	93.356076	0.187434	93.4	0.1904	94.1	0.1737	92.26	0.2178	93.76	0.186	63.45	0.6359	63.48	0.636	70.72	0.5649	62.5	0.6435	96.23	0.1109	94	0.1917
21	93.350895	0.188391	92.731741	0.20802	93.48	0.1883	93.74	0.1827	92.32	0.2158	93.89	0.1808	63.42	0.6358	63.89	0.6333	70.87	0.5643	62.89	0.6404	96.32	0.1092	94	0.1918
22	93.352459	0.188398	93.903738	0.176525	93.52	0.1869	93.97	0.178	92.3	0.216	93.59	0.1858	63.45	0.6359	63.76	0.6326	70.87	0.5648	63.62	0.6411	96.4	0.106	93.84	0.1967
23	93.317513	0.188262	93.608001	0.187275	93.58	0.1858	94.25	0.1716	89.54	0.268	93.17	0.1987	63.43	0.6357	63.37	0.6365	70.76	0.5652	56.56	0.7028	96.41	0.1057	93.69	0.1977
24	93.295607	0.188684	93.963198	0.17614	93.56	0.1856	94.22	0.1723	91.33	0.2365	93.18	0.1959	63.33	0.6359	63.84	0.6383	70.97	0.564	63.37	0.6425	96.47	0.1041	94.02	0.1966
25	93.357675	0.187943	93.753523	0.182915	93.6	0.185	94.17	0.1704	92.17	0.2194	93.78	0.1836	63.51	0.6357	63.12	0.6372	70.92	0.5642	62.9	0.6429	96.49	0.1023	94.01	0.1977
26	93.293521	0.188805	93.980411	0.175092	93.54	0.186	93.61	0.1896	92.34	0.2158	93.73	0.1872	63.43	0.6353	63.67	0.6351	70.9	0.5647	63.22	0.6439	96.57	0.1003	93.75	0.206
27	93.285176	0.188988	93.86149	0.17697	93.66	0.1836	94.32	0.1699	92.43	0.2136	93.81	0.1838	63.52	0.6352	63.53	0.6354	70.94	0.5632	57.59	0.6719	96.57	0.1	93.94	0.1941
28	93.35924	0.189168	93.944422	0.175131	93.67	0.1829	94.13	0.1765	92.44	0.2129	93.89	0.181	63.56	0.6354	63.77	0.6344	71.04	0.5626	60.17	0.6569	96.65	0.0987	93.86	0.2038
29	93.327424	0.188806	93.052514	0.199541	93.68	0.182	94.26	0.1709	92.46	0.2126	93.9	0.1799	63.49	0.6352	63.78	0.6336	71.07	0.5626	60.35	0.6562	96.66	0.0976	93.83	0.2066
30	93.2471	0.190023	94.104026	0.173485	93.6	0.1845	94.33	0.1688	92.6	0.2106	93.91	0.1802	63.54	0.6352	63.74	0.6354	71.02	0.5632	61.4	0.6522	96.78	0.0956	93.85	0.2049
31	93.2471	0.19004	94.063342	0.173329	93.72	0.1825	94.26	0.1683	92.6	0.2099	93.94	0.1797	63.41	0.6352	63.26	0.6364	71.11	0.5621	61.62	0.6499	96.75	0.0943	93.81	0.2091
32	93.230931	0.189578	93.595483	0.186601	93.72	0.1816	93.87	0.1804	92.65	0.2097	93.98	0.1794	63.47	0.6353	63.79	0.6329	71.1	0.5621	59.68	0.662	96.81	0.0944	93.72	0.2119
33	93.248665	0.190243	93.927209	0.175719	93.69	0.1817	94.39	0.1648	92.63	0.2084	93.94	0.1777	63.53	0.6351	63.57	0.6355	71	0.5627	62.25	0.6494	96.8	0.0941	93.98	0.2121
34	93.280481	0.189962	93.908433	0.17595	93.82	0.1802	94.22	0.1693	92.7	0.2081	93.98	0.1778	63.56	0.6349	63.02	0.6384	71.13	0.5624	61.91	0.6493	96.84	0.0913	93.78	0.2168
35	93.246057	0.189721	94.043001	0.174542	93.81	0.1803	94.38	0.1664	92.71	0.2077	94.03	0.1771	63.5	0.6349	62.47	0.638	71.14	0.5619	60.66	0.6552	96.88	0.0913	93.72	0.2161
36	93.172514	0.192022	93.950681	0.17568	93.77	0.1806	94.19	0.1711	92.78	0.2061	94.05	0.1759	63.49	0.635	63.46	0.6362	71.11	0.5614	61.61	0.653	96.91	0.0912	94.02	0.2082
37	93.249708	0.190662	93.517245	0.188194	93.78	0.1796	94.22	0.1701	92.71	0.2073	93.86	0.1847	63.81	0.6327	64.24	0.6302	71.2	0.5615	60.27	0.657	96.88	0.0909	93.87	0.2104
38	93.204331	0.191052	93.590788	0.183568	93.83	0.1793	94.38	0.1658	92.81	0.2058	94.1	0.1801	64.09	0.6309	64.14	0.6293	71.19	0.5613	60.86	0.6556	96.93	0.09	93.74	0.2104
39	93.188162	0.190624	93.939727	0.176433	93.87	0.1788	94.45	0.1657	92.81	0.2056	93.92	0.1773	64.15	0.6306	64.37	0.6287	71.15	0.5617	61.44	0.6498	96.96	0.0894	93.41	0.2265
40	93.195464	0.19164	93.789511	0.180775	93.85	0.1787	94.41	0.1642	92.83	0.2046	94.16	0.1746	64.06	0.6309	64.37	0.6288	71.13	0.5612	62.52	0.6506	96.98	0.088	93.77	0.2141