MACHINE LEARNING TOWARDS DATA WITH COMPLEX STRUCTURES

By

Runze Su

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Statistics — Doctor of Philosophy
Computational Mathematics, Science and Engineering — Dual Major

2022

ABSTRACT

MACHINE LEARNING TOWARDS DATA WITH COMPLEX STRUCTURES

By

Runze Su

The development of sequential analysis provides a deeper understanding in the exploration of many different fields. In the application of sequential analysis, there are two main challenges: How to extract informative features from a high-dimensional noisy domain? How to model the interaction for the information flow from multiple domains? We explored the two core challenges in bio-informatics, sales forecasting and multimedia services.

In biology field, a typical problem is the to evaluate the interaction mechanism between non-coding DNA sequences and transcription. We propose CANEE, a convolutional self-attention architecture to analyze the function of non-coding DNA sequences. Compared to other existing models, CANEE achieves a better performance in overall prediction of 919 regulatory functions with respect to receiver operating characteristics and has a significant improvement on some responses in precision recall curve with shorter training time. In sales forecasting field, we extract a unique customers' microbehavior dependency structure from clickstream data based on a Word-to-Vector model. Then,we build a clickstream informed LSTM model to forecast the car sales over 30 days. Our model significantly outperforms the classic seasonal autoregressive integrated moving average model. Besides, we demonstrate that transfer knowledge among different car models can further improve the performance. Other applications for multi-domain sequences happens in multimedia service field, where we focus on the understanding of multiple domain modalities. We propose new principles for audio visual learning and introduce a new framework as well as its training algorithm to set sight of videos' themes to facilitate AVC learning.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Overview

Data with complex structures extensively exists in lots of different fields. Thus how to extract informative features from a noisy domain and how to build a model based on those features is of great importance. Nowadays, with the rapid development of machine learning techniques, deep learning shows its tremendous potential in modeling the statistical relationships across multiple domains. In this paper, we showed the application of machine learning in bio-informatics, business sales forecasting and multimedia fields.

In bio-informatics field, evaluating the functional effects of non-coding DNA sequences has been an important and challenging problem. Although experimental results have indicated the connection between DNA sequences and gene expression, two main characteristics of the DNA sequence data is hindering the revealing of the interaction mechanism between non-coding DNA sequences and transcription. Firstly, the non-coding DNA sequences are longer than traditional sequential data. Secondly, the DNA sequences are hard to interpret and don't necessarily follow ordered properties. Several methods have been proposed using deep learning methods to capture the interaction, but the sequential learning mechanisms are mainly based on either a fully connected or a bidirectional long short-term memory frameworks, which requires a large memory and hence is difficult to scale to long DNA sequences. To address those challenges, we proposed a convolutional self-attention architecture to analyze the function of non-coding DNA sequences. Compared to other existing models, our model achieves a better performance in overall prediction of 919 regulatory functions with respect to receiver operating characteristics and has a significant improvement on some responses in precision recall curve with shorter training time. We also found out that there exists interactions for the functional analysis across different species. To look into the information sharing between human and plant genes, we proposed a model with similar structure in predicting plant stress response from DNA sequences. We demonstrate that our model outperforms classical shallow and deep learning approaches for predicting plant gene expression and it shows great potential with pretrained information from experiments on human genes.

In business sales forcasting field, we explore the clickstream data for car sales. Forecasting car sales and demand in the market is an important but challenging task in car market analysis. Recently, the development of deep learning and the availability of clickstream data with rich customers' online behaviors provide us with immense opportunities to advance the car demand forecasting problem. However, the online clickstream data is very noise and less informative. To solve the problem, we consider the clickstream data as a sentence consists of words, then we applied a word-to-vector model to extract a unique customers' micro-behavior dependency structure from clickstream data. Then, we build a clickstream informed LSTM model to forecast the car sales over 30 days. Our model significantly outperforms the classic seasonal autoregressive integrated moving average model. Besides, we demonstrated that transfer knowledge among different car models can further improve the performance.

In the multimedia field, we put our attention on short videos. Comparing with image, text and audio, the short videos are much more informative and more complicated. The applications of short-term user-generated video (UGV), such as Tik Tok, Snapchat and Youtube short-term videos, booms recently, raising lots of multimodal machine learning

tasks. Among those tasks, learning the correspondence between audio and visual information from videos is a popular but challenging one. Though there exist lots of classic methods to extract features from audio and visual domains, how to measure the correspondence across multiple domains remains a problem. Most previous work of the audio-visual correspondence(AVC) learning only investigated constrained videos or simple settings, which may not fit the application of UGV. For this problem, we proposed new principles for AVC and introduced a new framework to set sight of videos' themes to facilitate AVC learning. We also released the KWAI-AD-AudVis corpus which contained 85,432 short advertisement videos (around 913 hours) made by users. We evaluated our proposed approach on this corpus, and it was able to outperform the baseline by 23.15% absolute difference. Based on that, with a better understanding on the interaction for multimodality, we further explored a novel end-to-end self-organizing framework for user behavior prediction. The new model is able to learn the optimal topology of neural network architecture, as well as optimal weights, through training data. We evaluate our proposed method on our in-house dataset. The experimental results reveal that our model achieves the best performance in all our experiments.

Chapter 2

Machine learning on DNA sequences

The analysis of DNA sequences has been a challenging problem in both academic and industry field. Here we proposed a convolutional self-attention network to learn the internal effects for DNA sequences. The application of this model fells in human and plant DNA sequences. In section 2.1, we will cover the functional analysis on non-coding DNA sequences from human being with a convolutional self-attention network. In section 2.2, we apply the similar architecture on the genetic motif of plants. Beside the higher performance comparing with traditional models, we also did further analysis on the training speed and transfer learning potention. Comparing with the state-of-the-art models, the convolutional self attention network shows a faster training speed, reasonable interpretability and large potential for transfer learning across species.

2.1 Functional analysis on non-coding DNA sequences

Understanding the function of human DNA sequences has been an important but challenging problem. Experimental results have indicated that non-coding DNA sequences may act on the constitution of numerous diseases, but concrete functional evaluation of non coding DNA sequences remains a problem. One concerning point is to evaluate the relationship between DNA sequences and their corresponding transcription process. The core challenge for this problem is to evaluate the binding of chromatin proteins and histone marks from DNA sequence with single-nucleotide sensitivity. TF binding can be influenced by cofactor binding sequences, chromatin accessibility and structural flexibility of binding-site DNABenveniste *et al.* (2014); Whitaker *et al.* (2015). DNase I-hypersensitive sites (DHSs) and histone marks are expected to have even more complex underlying mechanisms involving multiple chromatin proteinsSlattery *et al.* (2014). To learn how non-coding DNA acts on those factors, a stable and accurate sequence tagging model is of great importance to reveal the complicated dependencies.

Deep learning has shown its strong potential in dealing with rich-feature problems from large data sets LeCun *et al.* (2015). Current genomics problems are based on DNA sequencing, which makes them to be rich-feature and large sample problems. DeepSEA Zhou and Troyanskaya (2015) is the first model for this problem. It proposed a deep convolutional neural network to evaluate the sequence tags. Then, DanQ Quang and Xie (2016) combines the bidirectional recurrent neural network architecture with the convolutional neural network to capture sequential properties of DNA sequences. Besides, it's hard to classify which weather a DNA sequence is the forward one or the reverse complementary sequence. To learn this property, DeFine Wang *et al.* (2018) and FactorNet Quang and Xie (2019) take both the forward and reverse complementary sequence as the model input.

In the recent years, the development of natural language processing has provided us with a new sight of sequence learning. Recurrent neural network Mikolov *et al.* (2010) captures the sequence properties by setting different units to scan the input sequence. Classic recurrent neural network frameworks including gated recurrent units Chung *et al.* (2014) and long short term memory Hochreiter and Schmidhuber (1997a). Bidirectional recurrent neural networks Schuster and Paliwal (1997) strengthen the learning ability by learning both the forward and reverse ordered sequence. A self-attention framework Vaswani *et al.* (2017) is proposed to learn a score for each element with respect to all other elements in the sequence. However, existing methods for the non-coding DNA functional evaluation always rely on an ordered recurrent neural network to learn the sequential properties of DNA sequences. This is counterintuitive since non-coding DNA sequences don't necessarily affect transcription by the order of base pairs. Especially in long DNA sequences, there are 2 main challenges:

- Recurrent neural network is likely to cause gradient vanishing problems in long DNA sequences.
- The interaction for the base pairs in the DNA sequence is complicated.

To better model the function of non-coding DNA sequences, we propose CANEE, a convolutional self-attention architecture to evaluate the transcription effects of non-coding DNA. This model applies a convolutional layer to convert DNA base pairs to a sequence of numerical values. Then we use a self attention module to learn the interaction between each base pair and all other base pairs in the DNA sequence. Because weights in self attention layer will be updated parallelly, it will also avoid the gradient vanishing problem. In summary, CANEE will achieve a higher prediction accuracy and faster training speed.

2.1.1 Task and dataset overview

The dataset we are using is collected in the paper of DeepSEA Zhou and Troyanskaya (2015). They collected the human GRCh37 reference genome and then segmented DNA into short sequences with a length of 1000. Different DNA sequences will not overlap for more than 200 base pairs. On each short sequence, they evaluate the 919 regulatory functions including factor binding, DNase I sensitivity and histone-mark effects. If more than 500 base pairs are activated with such functions, then they labeled the corresponding response as 1. We take the DNA sequence as a 1000×4 matrix, where the first dimension represents the sequence length and the second dimension represents the type of the base pairs A, T, C and G.

The dataset contains the forward DNA sequences and their complementary sequence pairs. The predicted probability for each sequence will be the average of the forward and reverse complementary sequences. The training set, testing set and validation set are already split. The training set contains 4400000 samples, the testing set contains 455024 samples an validation set contains 8000 samples. No reverse complementary is leaked into other sets.

2.1.2 Model formulation

Figure 2.1 illustrates the framework of CANEE model. The sequences are firstly converted to 1000×4 matrices, then the input matrix will go through a 1D convolutional layer along with a max pooling layer. The output sequence from the max pooling layer will then be fed into the next layer for positional embedding. In the end, a multihead self attention network and a fully connected layer will learn and provide the output. We choose binary cross entropy as the loss function. To avoid overfitting problem, we also add dropout layers in the self attention layers and an early stopper to keep the best result if the model doesn't receive a lower loss in validation set for 5 epochs.

The modules will be introduced here:

CNN Module

We apply a convolution operation to the input matrix. It consists of a 1D convolutional layer and a max-pooling layer. Suppose the input of the convolutional layer formula is (N, I, L) and the output is (N, O), then the 1D convolutional layer will as following:

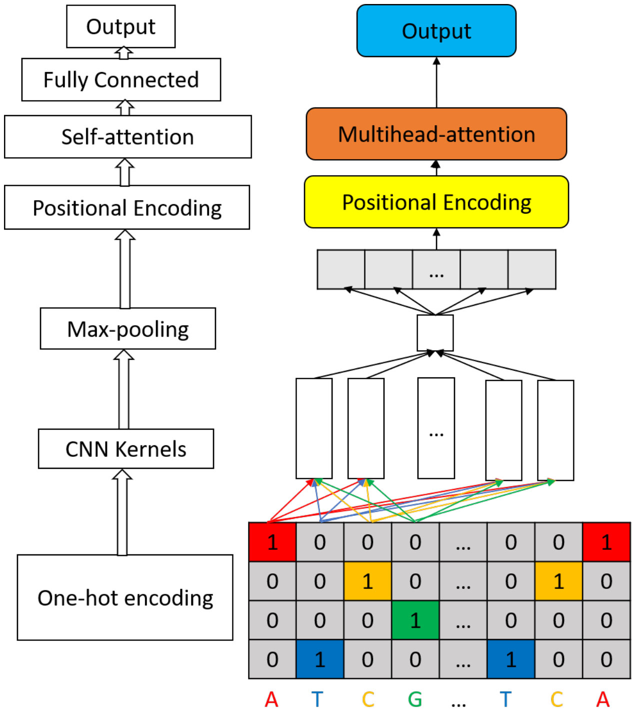


Figure 2.1: The architecture of CANEE model.

$$Conv1D(X_{N_m,O_j}) = ReLU(Bias(O_j) + \sum_{k=0}^{I-1} W_{O_j,k} \star X_{N_m,k}),$$

where N is the batch size, I is the dimension for elements in the input sequence, O represents the output element dimension, L is the input sequence length. Then the output sequence from the convolutional layer will be fed into a max-pooling layer:

$$Output(N_i, C_j, k) = \max_{m=0,1,\ldots,kernel\,size-1} input(N_i, C_j, k+m),$$

where input value is of size (N, C, L).

Positional Encoding

Self attention architecture updates the weight in parallel but will miss element location information. To add the relative position information to the model, we feed the output sequence to a positional embedding layer. Positional embedding encodes sines and cosines of different frequencies as positions. Assume t is the location for an element in the sequence, then its positional encoding $p(t)$ is defined as:

$$p(t) = \begin{cases} \sin(\frac{t}{10000^{2k/d}}) \ if\, t = 2k \\ \cos(\frac{t}{10000^{2k/d}}) \ if\, t = 2k+1 \end{cases}$$

Self Attention Module

This module consists of a self attention network and a fully connected layer. The self attention network has three factors: query Q, key K and value V. Assume the input is X, the formulation can be expressed as:

$$Q_i = W_Q X_i$$

$$K_i = W_K X_i$$

$$V_i = W_K X_i$$

$$S_{i,j} = \frac{Q_i \cdot K_j}{\sqrt{d}}$$

$$Score_{i,j} = \frac{exp(S_{i,j})}{\sum_k exp(S_{i,k})}$$

$$output_i = \sum_j S_{i,j} V_j$$

, where W represent the weight matrix. The output from self attention network will be fed into a fully connected neural network to fit the 919 tags.

We pick up DeepSEA and DanQ as our comparison. The model architecture is summarized in Figure 2.2. The major difference for the the two models and CANEE is

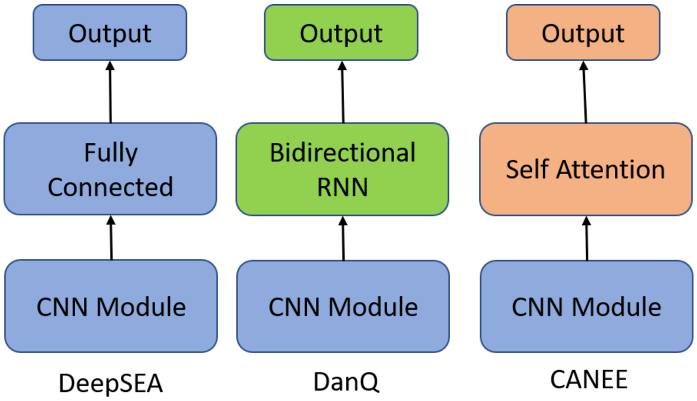


Figure 2.2: Model structure comparison.

Models	**AUC-ROC**	**AUC-PR**
DeepSEA	0.9325	0.3425
DanQ	0.9384	0.3709
CANEE	**0.9398**	**0.3732**

Table 2.1: Model performance comparison.

the sequence learning module. CANEE applies self attention module while the DanQ and DeepSEA applies Bidirectional RNN and simple fully connected layer.

2.1.3 Experimental results

In the experiments for CANEE, we select kernel stride as 26 and kernel size as 75. We set 4 heads in the multi-head self attention layer and stack 2 self attention layers in the self attention module. The learning rate we set is 0.0001. Weights are initialized by Xavier uniform. All experiments are conducted on single NVIDIA V100 GPU.

Performance Analysis

We compared the performance of CANEE with DeepSEA and DanQ. Two metrics are used to evaluate model performance: Receiver Operating Characteristics(ROC) and Precision Recall Curve(PR). The area under receiver operating characteristics curve(AUC-ROC) will provide an estimation of how the model learn negative samples while the area

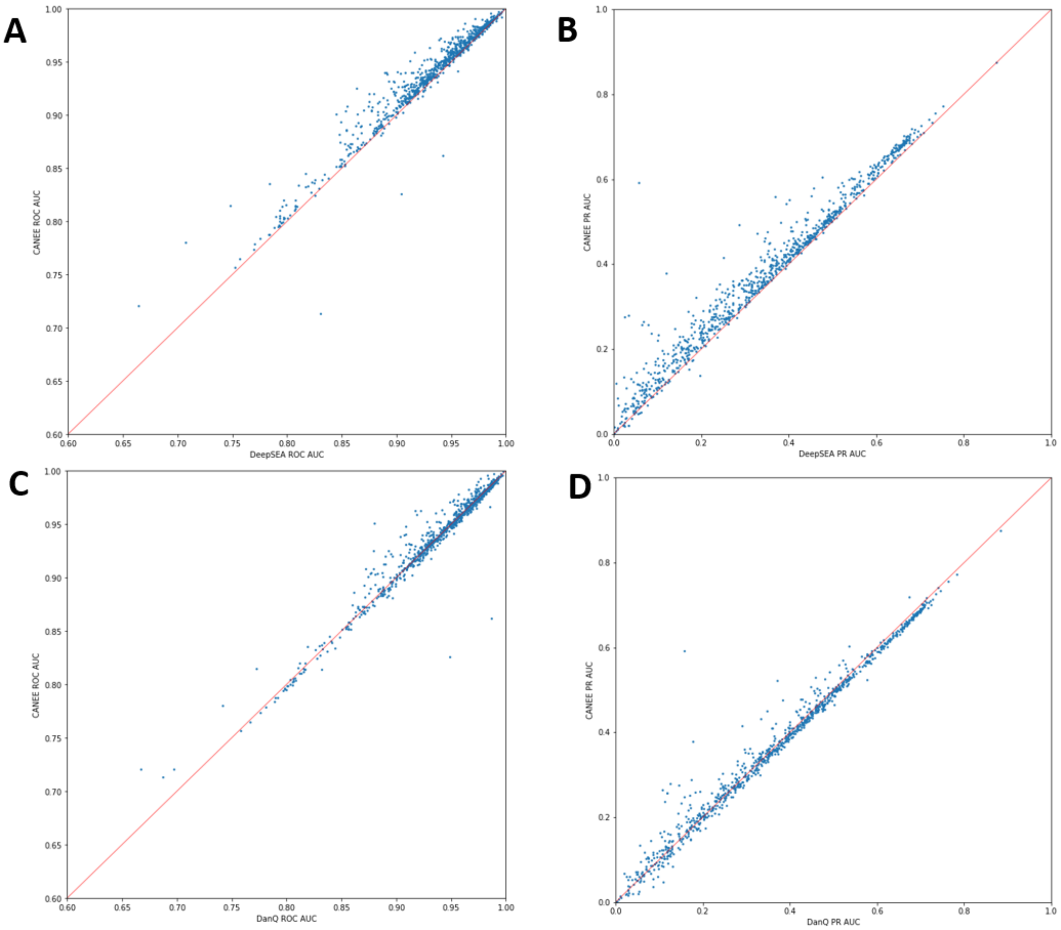


Figure 2.3: Figure A: CANEE output vs DeepSEA in AUC-ROC. Figure B: CANEE output vs DeepSEA in AUC-PR. Figure C: CANEE output vs DanQ in AUC-ROC. Figure D: CANEE output vs DanQ in AUC-PR.

under the precision recall curve(AUC-PR) will focus more on the model performance on positive samples. In the dataset, positive samples are significantly less than negative samples, so we expect AUC-ROC to be much higher than AUC-PR.

We analyze our model performance in two aspects. First we evaluate the overall accuracy for all regulatory functions, then we check the distribution for each response prediction. In Table 2.1, we calculate the average AUC-ROC and AUC-PR for all 919 responses in testing set. For both metrics, CANEE outperforms both DeepSEA and DanQ.

Another important point is to find out which responses have significant improvement.

Cell Type	TF, DNase or HistoneMark	DanQ AUC-PR	CANEE AUC-PR	AUC Difference
GM12878	BRCA1	0.156832	0.592497	0.435665
HepG2	BRCA1	0.176535	0.377814	0.201279
GM12878	ZBTB33	0.107438	0.264291	0.156852
GM12878	ZZZ3	0.125252	0.278512	0.15326
H1-hESC	NRSF	0.370143	0.521568	0.151425
HepG2	ZBTB33	0.117476	0.257853	0.140377
K562	ZBTB33	0.117976	0.257172	0.139195
K562	BDP1	0.144466	0.276138	0.131672
K562	CHD2	0.290002	0.415569	0.125567
H1-hESC	BRCA1	0.113197	0.237324	0.124127

Table 2.2: Top 10 regulatory factors with the highest PR-AUC improvement.

Figure 2.3 shows the comparison between DeepSEA, DanQ and CANEE model by each response. As a consequence, CANEE outperforms DeepSEA in most responses in AUC-ROC and AUC-PR. Besides, CANEE shows a similar performance as DanQ in most of responses, but it can also be noticed that some responses show a significant better performance in CANEE, especially with respect to AUC-PR. Taking 5% as a threshold, more than 95% of responses show no significant difference between DanQ and CANEE, but in 40 responses which present significant differences, 38 of them show a better AUC-PR in CANEE model. Table 2.2 lists the top 10 regulatory functions that have the highest improvement. The highest improvements all happen in cell types such as GM12878, HepG2, H1-hESC and K562, which indicates that the improvement is closely related to cell type. Besides, taking GM12875 for example, among 91 regulatory functions of this cell type, the AUC-PR can increase 4.2% by switching from DanQ to CANEE.

Speed Comparison

Comparing with recurrent neural network model, self attention is much faster in training. Under the same environment, it takes 30 $\sim$ 60 epochs until convergence while CANEE takes

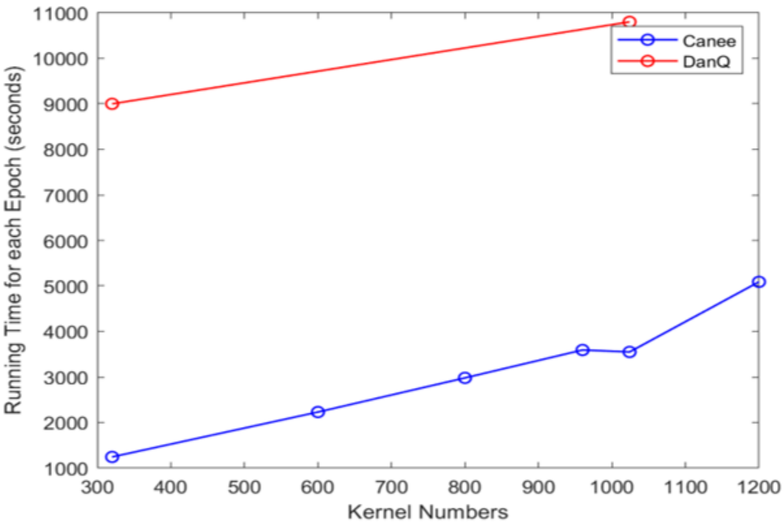


Figure 2.4: Running speed: DanQ vs CANEE.

$30 \sim 50$ epochs to converge, but CANEE is much faster for each epoch. We train the model with different CNN kernel numbers and record the run time in Figure 2.4. It turns out that under the same setting, CANEE is much faster in this long sequence learning.

2.1.4 Discussion

In this section, we propose a new architecture to evaluate the transcription effects of non-coding DNA. The model shows a better performance in AUC-ROC and AUC-PR while accelerating training procedure within less training time.

However, there are still challenges remaining in this problem. Here we will list some future directions to further update the model. According to the biological interpretation of the transcription results, there exists interactions between 919 targets and this still needs exploration; a deeper understanding on how to combine the forward and reverse complementary sequences may further improve model performance; taking sparsity in the self attention module into consideration can also help the model better capture the information behind the DNA features. It is also valuable to find a lower-dimensional representation for the interaction between base pairs, and this may make the model even faster and cut down noise factors in genomics field.

2.2 The prediction of plant stress response from DNA sequences

Advances in omics technologies have led to an abundance of biological information. Integrating rich data sources from this omics data explosion, biologists can get a deeper understanding of complex biological systems and answer difficult questions by employing deep learning, which enables successful prediction by extracting high-level features from massive data Zhou and Troyanskaya (2015); Quang and Xie (2016). A central problem in bioinformatics towards understanding these complexities is gene function prediction, in particular molecular functions (e.g. transcription factor binding) and biological processes (e.g. a given gene is pertinent to the process of reproduction). However, experimentally annotating gene function is a relatively slow process Kulmanov *et al.* (2018), making attractive these computational methods on DNA sequence data.

A subclass of this central problem is found in molecular plant biology, where much research is done to understand how plants respond to various abiotic and biotic stressors (e.g. heat waves, drought, and pest infestations). As the regulation of expression levels determines how plants respond to different environmental factors, the analysis of expression regulation is of great importance.

A main component of gene expression regulation is through the binding of transcription factors to specific sequences of DNA called regulatory elements (motifs). For this reason, an avenue of research has been to identify these transcription factors and the respective regulatory motifs, in order to predict gene expression responses Uygun *et al.* (2017); Wilkins *et al.* (2016). However, identifying individual regulatory motifs, such as transcription factor binding sites (TFBS), is only a small part of the complex process of gene regulation. Indeed, gene regulation processes also depend on the location, orientation, quantity and co-localization of regulatory motifs. These dependencies form the structures that modulate gene regulation, and these structures form what is called regulatory grammar Weingarten-Gabbay and Segal (2014).

Understanding of regulatory grammar by computational modeling of these complex dependencies has thus become a hot area of bioinformatics research. Many advancements towards modeling complex regulatory grammar have come from deep sequence learning models, traditionally used in natural language processing.

One of the early deep learning models developed to account for the sequential dependencies was DeepSea Zhou and Troyanskaya (2015). This was done by using convolutional neural networks (CNN), from which motifs and local dependencies were learned, ultimately used for functional-variant prediction.

Building on the DeepSea model, Quang and Xie developed DanQ Quang and Xie (2016), which couples the CNN with a recurrent neural network (RNN), namely a bi-directional long short-term memory network (LSTM) Hochreiter and Schmidhuber (1997b). The LSTM component helps identify long-range dependencies [9], and hence co-localization dependencies. As the LSTM is bi-directional, it learns these features on both the forward and reverse ordering of sequences (hence orientation). Besides, as discussed in the previous topic, the self-attention module points out a promising direction towards the understanding of genetic information.

These developments of deep sequence learning models are easily tailored and applied to

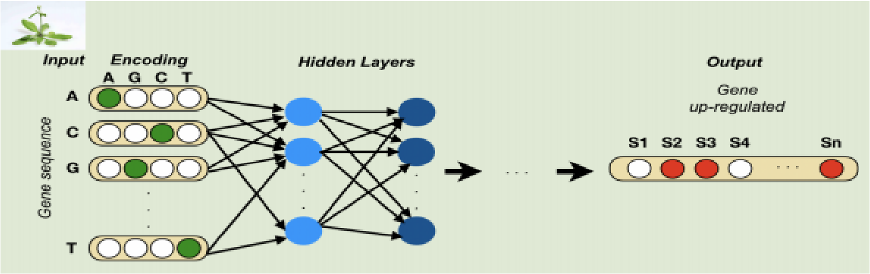


Figure 2.5: High level pipeline of DeepCAT.

our problem of interest: predicting plant stress response from DNA sequences. Building on these ideas, we propose DeepCAT, a similar convolutional self-attention architecture as CANEE to predict plant stress response from DNA sequences. DeepCAT consists of 3 layers. The first is a convolutional layer which converts DNA base-pairs to a numerical sequence, identifying key predictive motifs and local dependencies. The second layer is self-attention, which captures key predictive co-localization dependencies. Lastly, a fully-connected (FC) layer to output prediction scores of gene up-regulation under different abiotic and biotic stresses.

2.2.1 Data and problem statement

Gene expression and sequence data of 20,799 Arabidopsis genes each consisting of 3,200-bp (covering promoter and 5' UTR) were downloaded from the AtGenExpress database and processed as in Uygun *et al.* (2017). In brief, the preprocessed and normalized expression data from AtGenExpress was used to calculate log2 fold change between stress and control conditions using Limma Ritchie *et al.* (2015) in the R environment. Genes with a log2 fold change ≥ 1 were considered up-regulated.

DNA sequences were pulled for each gene from TAIR10. Particularly, the sequences were taken from 1-kilobase (kb) upstream and 500-base pairs (bp) downstream the transcription start site and 500-bp upstream and 1-kb downstream the transcription stop site. These sequences were then one-hot encoded, with each sequence converted into a 3200x4 binary matrix. The columns correspond to A,C,G,T, and rows correspond to the position in the DNA sequence, with a each row containing a single 1 in one column and zeros in the remaining columns.

Given raw DNA sequence data, the objective is to predict the gene expression responses to 57 environmental stress conditions in arabidopsis thaliana. Specifically, we want to predict if an arabidopsis gene was up-regulated or not in shoot tissue under each of 36 abiotic (e.g. cold, heat, osmotic) and 21 biotic (e.g. 71 Pseudomonas syringae, bacterial flagellin) stress conditions. Genes were randomly assigned according to a training-validation-test split of 70-10-20.

2.2.2 Model Architecture

As previously described in brief, DeepCAT consists of 3 main modules: (1) CNN, (2) Self Attention and (3) FC & output. The descriptions are below.

CNN Module and Self Attention Module

The CNN module and self attention module have the same structure as in Section 2.1.2.

Fully-Connected Output Module

The output of the self-attention module is the input here. We apply a single FC layer, giving weighted scores for each of the 57 stress types. We then apply a sigmoid output layer, which takes these scores and converts them to probability scores, which is a predicted probability of gene up-regulation under each of the 57 stresses.

Training

We trained DeepCAT by minimizing the average multi-task binary cross-entropy loss in

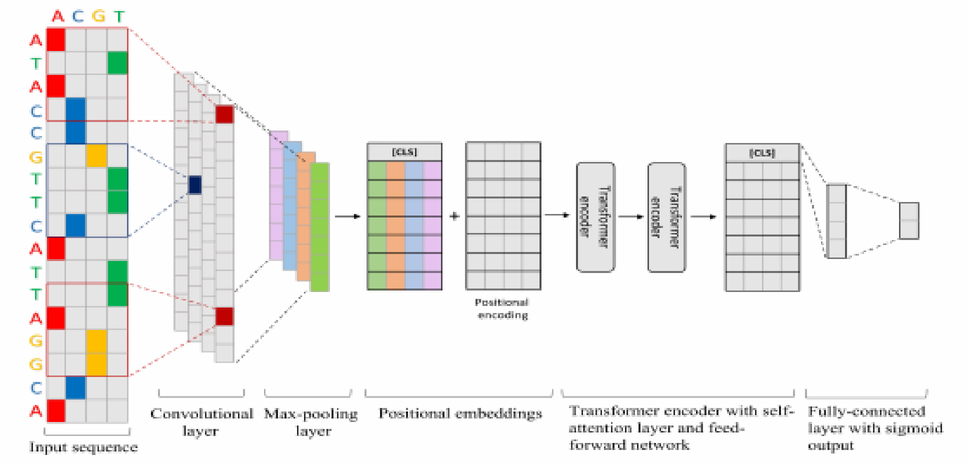


Figure 2.6: DeepCAT architecture.

mini-batches of size 50 using the Adam optimizer Kingma and Ba (2014). All the weights and biases were initialized with Xavier (uniform) Glorot and Bengio (2010) and zero values respectively. For model regularization purposes, we applied dropout with rate of 0.1 in attention layers.

Validation data was used to determine an optimal number of training iterations. Namely, we use an early-stopper to stop the training process if the validation loss does not decrease for a set number of epochs (default 5), thus keeping the model that performs best on the validation set.

In all of our experiments we trained DeepCAT with settings: 320 convolutional kernels/filters, kernel dimension 26, pooling dimension 13, and used 4 attention heads. Our implementation was with PyTorch, and our experiments (training and testing) were ran on NVIDIA K80 GPU.

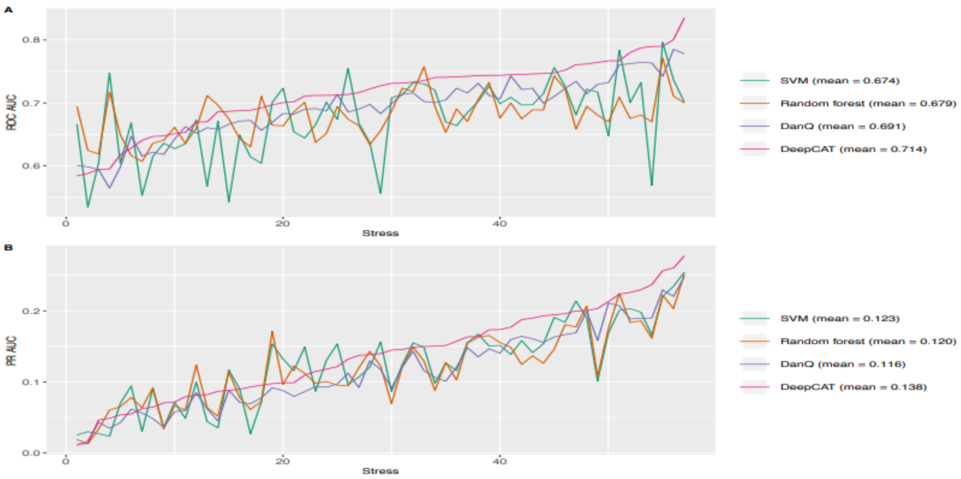


Figure 2.7: Performance of DeepCAT.

2.2.3 Experimental results

2.2.3.1 Performance Analysis

Using the fully trained models, performance was measured on the testing data. We used two metrics: the Receiver Operating Characteristic-Area Under the Curve (ROC-AUC) and the Precision Recall-Area Under the Curve (PR-AUC). For overall comparison purposes we averaged the PR-AUC and ROC-AUC across the 57 stress types.

Experiment 1

In the first experiment, we evaluated baseline performances of our standard DeepCAT model and a few classic and deep learning models. The baseline models consisted of Support Vector Machines (SVM) and Random Forest. The deep learning model we compared against was essentially the DanQ model Quang and Xie (2016), with the modification of the output layer to give plant response probability scores for the 57 different stress types. We chose this deep learning model for comparison, as it has a similar structure as DeepCAT, and has performed well on a different but similar problem with human DNA data. Figure 2.7 shows the PR-AUC and ROC-AUC values (y-axis) for each of the 57 stress conditions (x-axis),

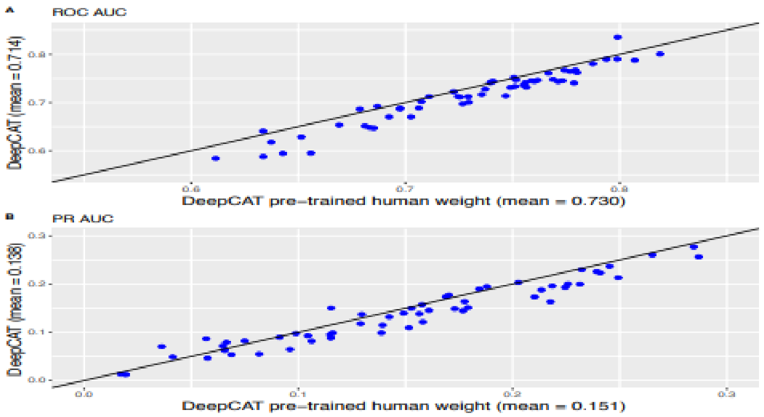


Figure 2.8: Performance of DeepCAT with kernels initialized from weights learned from the DanQ human model.

along with the respective average values in the legend. Comparing with the classic models and DanQ, DeepCAT can achieve a higher accuracy in most targets in average.

Experiment 2 - Transfer Learning

The main idea of Transfer Learning Tan *et al.* (2018) is to leverage existing knowledge from one problem to solve a different but similar problem. Here we injected existing existing knowledge in two ways. One was through experimentally verified information. The other was information learned from a model with a rich data set. As the kernels in the CNN layer of DeepCAT act as DNA motif finders, we experimented initializing the kernels with known A.thaliana TFBMs. Moreover, the DanQ model used a massive amount of human gene data (> 4 million), so we also experimented initializing the kernels with the kernel weights learned in the DanQ model. According to Figure 2.8 and Figure 2.9, we find that implementing these Transfer Learning methods in DeepCAT lead to better performances across nearly all 57 stresses.

Experiment 3 - Stress Grouping

Our previous results are based on learning all 57 stress responses simultaneously. However, the information from DNA sequences may not be shareable for different

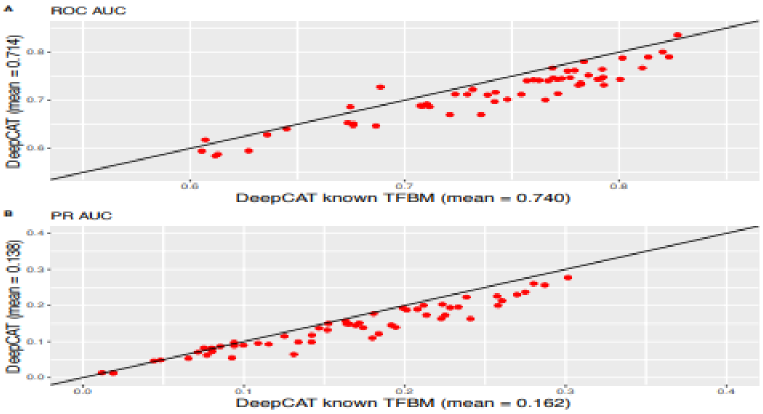


Figure 2.9: Performance of DeepCAT with kernels initialized from experimentally verified TFBMs.

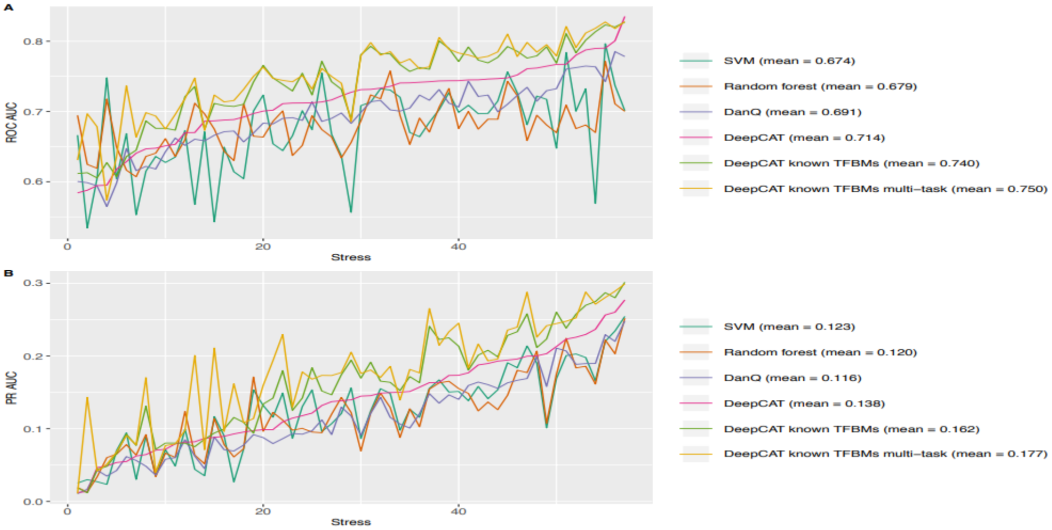


Figure 2.10: Performance of DeepCAT with known TFBM initialized kernels, and the clustered response multi-task model.

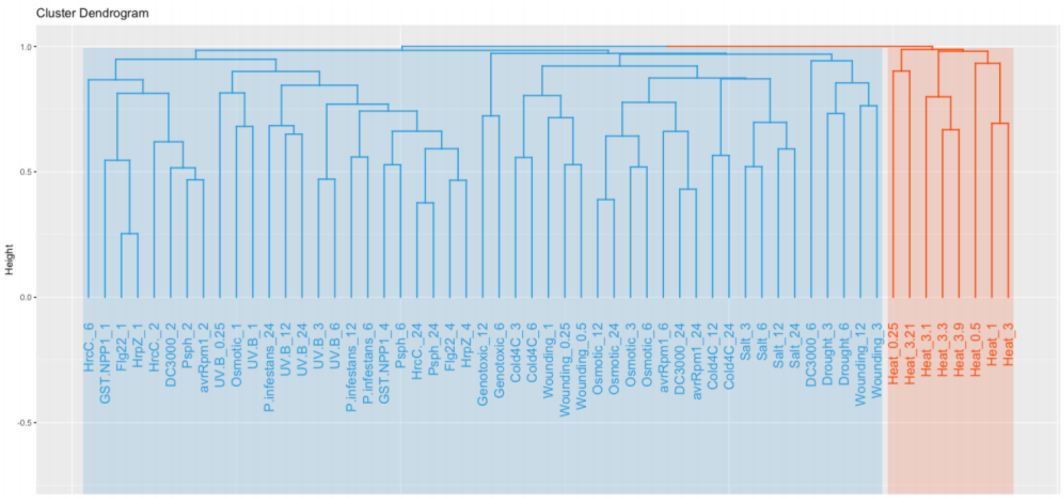


Figure 2.11: The clustering hierarchy of the stress types from k-means clustering, with red highlighted stresses being heat related.

responses, because these different stress types may have very different underlying regulatory mechanisms, and finding a good shared representation may not be possible. The expectation is that in an MTL setting, learning stresses with similar underlying regulatory mechanisms are to mutually benefit from each other, while stresses with very different underlying regulatory mechanisms may hinder performance.

In Figure 2.11, we did hierarchical clustering and cluster the responses into 3 different groups. Then, we train three models individually on those three groups. We also pair this with what we did in experiment 2 by initializing the convolutional kernels with known A.thaliana TFBMs. As is shown in Figure 2.10, we find that both of these experiments lead to better performances, with the latter yielding the best performance.

2.2.3.2 Interpretation

We also explored the interpretation of the convolutional and self attention layers in the DeepCAT model. An interesting result is that the DeepCat model can be interpreted as a

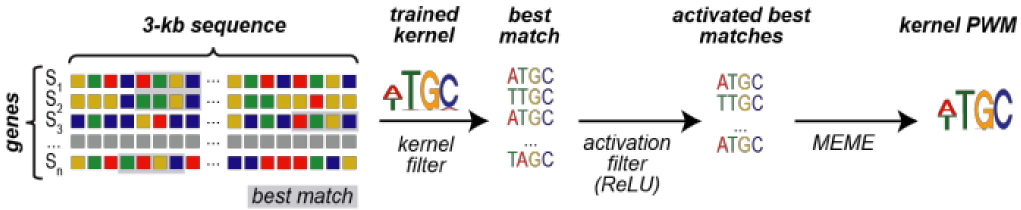


Figure 2.12: Pipeline to translate kernel weights to position frequency matrices and aligned to known motifs.

motif learner, by a translation of the kernels in the convolution layer to positional weight matrices Alipanahi *et al.* (2015). We aligned these to known motifs from the DAP-seq and CIS-BP databases using TOMTOM software (`https://meme-suite.org/meme/tools/tomtom`). Of the 319 motifs learned by our model, 114 significantly match known motifs ($E < 0.1$); a threshold of 0.05 was used for p-value to measure the similarities. Figure 2.12 shows this process. In Figure 2.13, it could be seen that the trained convolutional kernels can be interpreted to be matching with some the existing gene motifs.

Besides, we also analyzed the attention scores in the self attention module, we found interactions of motifs exists at different positions. From Figure 2.14 we can see that the attention model identifies interactions between base-pairs at long ranges, and thus identifies long-range co-localization dependencies.

2.2.4 Discussion

While the performance of DeepCAT is good relative to well-established shallow and deep learning methods, the accuracy is still low in absolute terms. Thus there are still some challenges to overcome. From our results, we see that the stress grouping is a significant matter, and more sophisticated methods to learn the best groupings (i.e. the most related stresses) could help increase testing accuracy. Additionally, we saw leveraging transfer

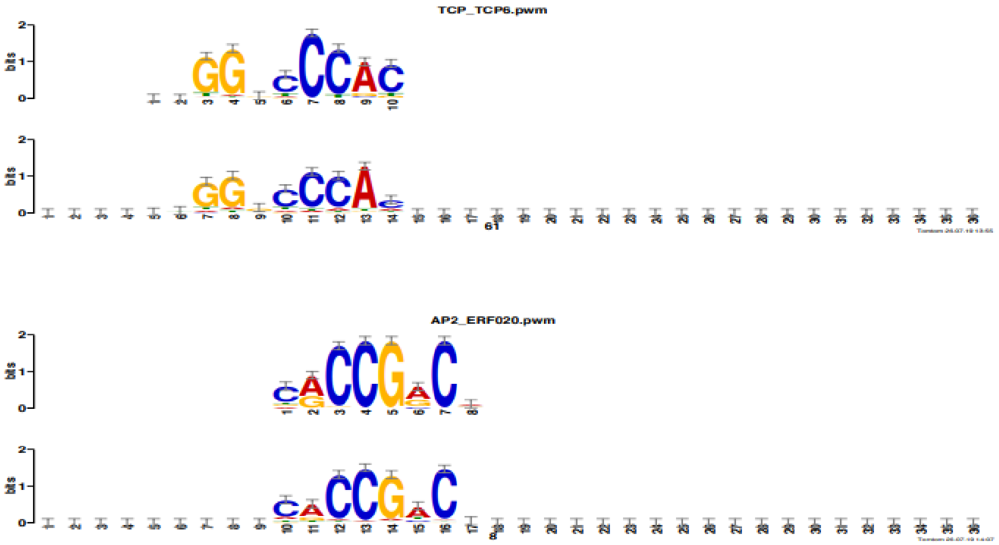


Figure 2.13: Motifs learned from DeepCAT aligned with a known TFBMs.

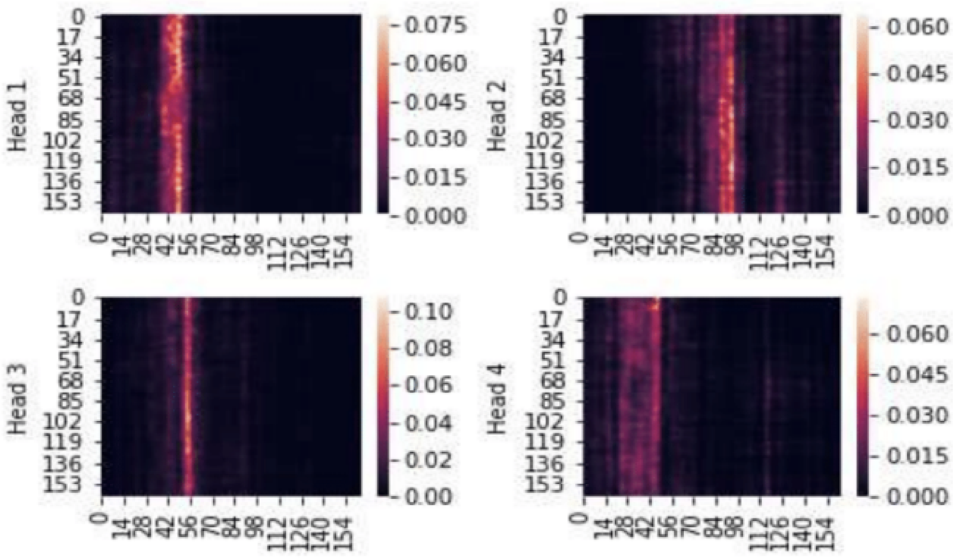


Figure 2.14: Here we plotted the motif interaction for the 4 heads of the attention module across all responses.

learning from both the big-data human model, and the experimentally verified TFBMs helped increase the predictive accuracy. This is another lever for increased accuracy, and hence is a direction of great interest.

Nonetheless, with DeepCAT we have shown how deep sequence learning and other learning mechanisms, such as grouped learning and transfer learning, can move us towards solving the problem of plant stress response prediction. Moreover, these methods are able to learn and extract key motifs and long-range motif interactions, which are important components of understanding regulatory grammar, and hence gene regulation.

Chapter 3

The understanding of clickstream data

3.1 Introduction

In recent years, car buyers are more demanding than ever: they emphasize various factors such as product variety, rapid delivery, etc. Meanwhile, the car-shopping behaviors change dramatically: customers spend, on average, 108 days in the market before buying a new vehicle and use 60% of their research time online before going to a dealership Cox (2018). This long time gap, typically 108 days, between the first click and orders provides car companies a unique opportunity to adjust their procurement and production. Therefore, a demand model based on clickstream data and local dealer information can help us seize these opportunities by making accurate predictions for the dynamic and granular automotive demand.

The past few years have witnessed a surge of interest in developing demand forecasting models utilizing low dimensional inputs such as dealers' locations, gas prices, gross domestic product, etc. Chase (2013). However, recent Internet clickstream tracking technology has generated massive data regarding customers' browsing behaviors. Huang and Van Mieghem (2014) suggested that forecasting models using the clickstream information can reduce the inventory holding by 3% and back-ordering cost by 5% in the rolling door industry. Those methods rely on the fact that online users and buyers are identical. However, in the automotive industry, customers tend to browse online for information but purchase vehicles

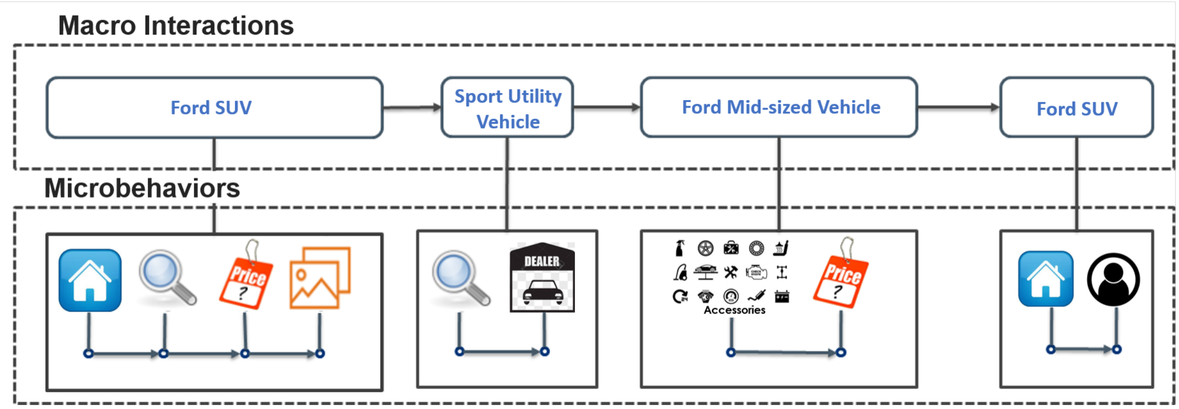


Figure 3.1: A example of a user from an e-commerce site.

in local dealers, which generates mismatches between the clicks and sales data. In summary, in auto demand forecasting, the proposed machine learning algorithms are desired to 1) ingest the high-volume and high-frequency clickstream data; 2) make no model assumption on the relationship between predictors and response variables; 3) accurately predict demands for various auto models (F150, Fiesta, Mustang, etc.) and entities (engine, drive, etc.); and 4) provide a robust long-term forecast. This study aims to develop novel machine learning methods to address the obstacles mentioned above.

Forecasting automobile demand has been studied for more than 30 years, pioneered by Lewandowski (1974) in the 1970s using conventional time series forecasting techniques. Then, Berkovec (1985) proposed a general equilibrium model that assumes that the demand equals the supply. In Brühl *et al.* (2009), moving average and Support Vector Regression are integrated with a Gaussian kernel for demand forecasting. In Kayapinar Kaya and Yildirim (2020), an 8-layer Deep Neural Network is developed to incorporate exogenous features such as exchange rates, gross domestic product, and consumer confidence indexes for sales prediction. However, all the above methods are based on monthly or quarterly data and are thus less applicable to process large data sets and provide accurate daily demand prediction. More importantly, none of the models leverage the massive clickstream data.

On the other hand, utilizing clickstream data for purchase forecasting has been studied extensively in e-commerce and has shown the ability to boost revenue significantly Xu *et al.* (2015); Lu *et al.* (2014). Precisely, a user's clickstream data consists of a series of visited items, known as macro interactions, and the final purchase, as shown in Figure 3.1. The dynamic and nonlinear temporal nature of clickstream data poses significant challenges to the existing time series forecasting models. These methods rely on the stationary and linear assumptions on the data, which is invalid in most cases. Therefore, numerous types of nonlinear Recurrent Neural Network (RNN) models, such as Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), and Transformer have been introduced Hochreiter and Schmidhuber (1997a); Vaswani *et al.* (2017); Cho *et al.* (2014). Zhou *et al.* (2018) further decompose each macro interaction into a sequence of micro behaviors and proposed an RNN based predictive model. Despite their successes in e-commerce, those methods require correspondence between online clicks and purchases. This critical requirement is natural for e-commerce but mostly missing in our study. The deeper understanding of transfer learning in Weiss *et al.* (2016) and graph neural network in Veličković *et al.* (2017); Zhou *et al.* (2020) points us a direction to learn from the clickstream data.

This study investigates automobile demand forecasting on a granular level via integrating historical sales and clickstream data. Although the clickstream data contains rich information, online users are anonymous and do not directly correspond to offline purchase data. This data nature poses tremendous challenges, including (a) how to extract useful features from clickstream data and (b) how to incorporate those features into the forecasting models. To address the challenges, we propose a general **C**lickstream based **D**emand **F**orcasting framework (CDF), which includes clicks and sales data into an RNN model for demand forecasting. Specifically, in the first step, we convert the clicks data into a sequence of micro behaviors. Then we leverage a natural language processing framework Word2vec in Mikolov *et al.* (2013), to learn the low dimensional numerical representation of the micro behaviors and cluster them into different groups. With the clustering structure, we transform the clickstream data into a multivariate time series, which is then fed into an RNN model to predict the demand. Our models outperform traditional machine learning methods in both national and state levels forecasting. Another contribution of our study is the application of transfer learning in auto demand forecasting. We demonstrated transferring knowledge across car models can also increase forecasting accuracy, especially for models with low sales. Besides, we also applied graph attention network (GAT) the learn the interaction between different car models.

The rest of this chapter is organized as follows. In Section 3.2, we give a formal illustration of the demand forecasting problem, and then Section 3.3 describes the data preprocessing and the proposed forecasting model. Detailed analysis and experiments are presented in Section 3.4. Finally, Section 3.5 discusses the potential extensions and future directions.

3.2 Problem Statement and Formulations

Let $\mathbf{M} = \{m_1, m_2, \ldots, m_M\}$ be the set of products, where M is the number of total models; $\mathbf{A} = \{a_1, a_2, \ldots, a_A\}$ be the set of customer actions on the web-page (e.g. accessory, gallery, Search inventory, etc.), where A is the number of possible actions; $\mathbf{D} = \{d_1, d_2, \ldots, d_5\}$ be the set of five discredited action dwell times corresponding to 0-20 percentile, 20-40 percentile, $\ldots$, 80-100 percentile respectively; $\mathbf{S} = \{s_{i,m}\}$ be the model m sales on the ith day. With these definitions, we can represent the clickstream data for a user as a sequence of tuples

Figure 3.2: An illustration of micro behavior.

(m_i, a_j, d_k), denoted as micro behaviors. Specifically, the micro behavior is defined as:

$$\text{Micro behavior} = (\text{Model, Action, Dwell Time})$$

As Figure 3.2 shows, elements in a micro behaviors are defined as:

- *Model* consists of two components: the car model and the release year. For simplicity, we only consider two types of release years: **the New** which is the current model when the customer visits the website and **the Old** is the older version.
- *Action* is certain short strings summarizing website content. For example, "fv:si" represents Ford vehicle search inventory.
- *Dwell time* represents the time customers spend on the website. We convert the duration into 5 categories including short, medium short medium, medium long and long according to the percentile of the duration time in all visits to reduce the dimensionality.

The problem we want to study is: *Given the historical sales for multiple car models and the historical clickstream data of a set of users, we aim to build a demand forecasting machine learning algorithm for different car models and entities.* Specifically, given sale data $\{s_{i,m}\}_{i=a-d}^{a}$ and clickstream data $\{c_{i,m}\}_{i=a-d}^{a}$ from day $a - d$ to day a for model m, we want to predict the model m demand on the day $a + l$. Here, d is the size of the look-back window, and l is the number of time steps we want to predict into the future. We then assume the following non-parametric model.

$$Y_{a+l,m} = f(\{s_{i,m}\}_{i=a-d}^{a}, \{c_{i,m}\}_{i=a-d}^{a}) + \epsilon,$$

where Y_{a+l} is the demand on day $a + l$ and $f(\cdot)$ is a nonlinear function, and ϵ is the random error.

Online Clickstream Data

We utilized two different data sources: Ford in-home historical sale data and clickstream data pulled from the Ford.com. Historical sales data contains daily updated customer purchases reported from dealers starting from Jan 6th, 2016. This sale data includes locations, dates, and model type. The unit of clickstream data is a web visitor who visits Ford.com. These data sets contain features for the visitors, including the IP address, date, browsing behavior, dwell time for each behavior, model, etc. However, web visitors are anonymous because they do not provide their identity. Thus, unlike e-commerce or the B2B cases Zhou *et al.* (2018), the web visitors in our clickstream data do not match the offline sales data. The original data contains 2,595,072,202 records (each record is an action for one visitor) with 977 features.

3.3 Statistical Analysis and Models

In this section, we introduce the statistical analysis and our model architecture. Section 3.3.1 will cover the data preprocessing and analysis and Section 3.3.2 will introduce the model we proposed.

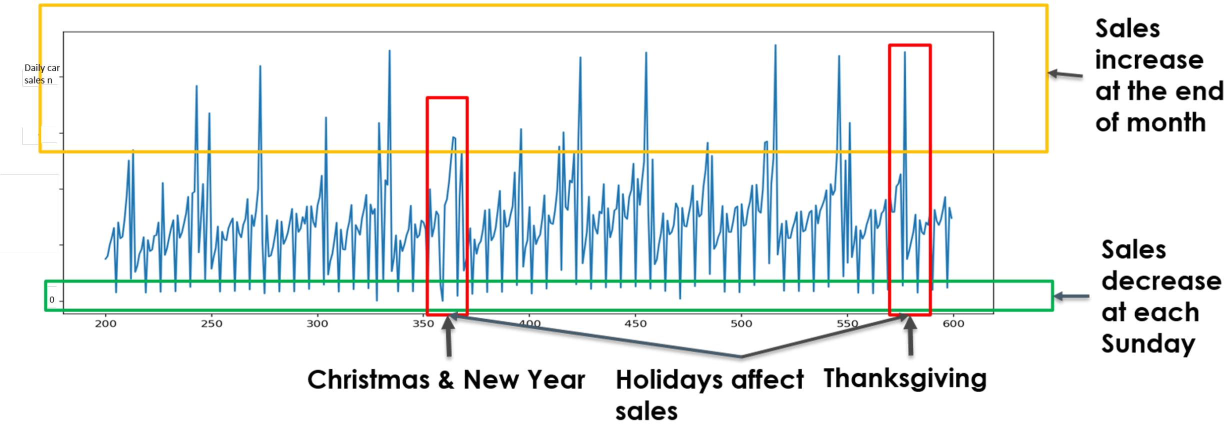


Figure 3.3: Sales demonstrate strong weekly, monthly, and holiday effects.

3.3.1 Data Preprocessing

The distribution of car sales is affected by compound temporal effects. To explore the factors in car sales, in Section 3.3.1.1 we figure out the temporal properties of the sale distribution. Then in Section 3.3.1.2, we analyze the correspondence between the online clickstream data and the sales history. As the clickstream data is not numerical, Section 3.3.1.3 introduces the embedding method and shows the properties of the clickstream data embeddings.

3.3.1.1 Historical Sales Data

To simplify this project and remove unnecessary heterogeneity, we only focus on the US's data. Besides, the sales data contains retail, leasing, and bulk orders by enterprises. The clickstream data track individual customers' browsing behavior, which is most related to retail sales. Thus, we focus on retail sales in this study.

There exist strong weekly effects and monthly effects on sales. As Fig 3.3 shows, there is a peak at the end of each month and a significant decrease every Sunday. National holidays also cause a compound effect on car sales.

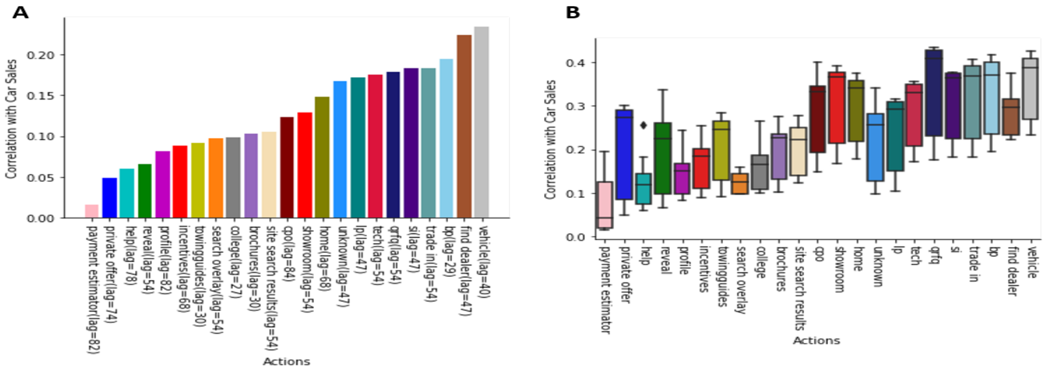


Figure 3.4: Fig A: The highest correlation between each micobehavior action and historical sales for one Ford model. Fig B: The correlation box plot between each microbahavior action and historical sales for Ford SUV.

3.3.1.2 Correlation Analysis between Clickstream data and Future Demand

There are hidden inferences between car sales and clickstream data. Figure 3.4 shows the correlation between Ford SUV microbehavior actions and Ford SUV sale history. Figure 3.4.A shows the ordered correlation between Ford SUV sales and each action in microbehaviors. Actions such as 'payment estimator', 'private' and 'help' are very sparse in the microbehaviors, and their low correlation with sales is within our expectation. Popular and intuitively related actions such as 'bp', 'find a dealer' and 'vehicle' show a relatively high correlation with historical sales. As there exist significant week effects in car sales, we also analyze the correlation between actions and car sales on different days of the week. Figure 3.4.B shows that those highly correlated actions with sales also show their high correlation on all days of a week.

Due to the mismatch between sales and clickstream data, we utilize an unsupervised Word2vec model, i.e., Continuous Bag of Words Model (CBOW), to embed micro behaviors into low dimensional vectors so that we can use machine learning algorithms to learn the relationship between micro behavior and extract features for the CDF model. Thus, we utilize the hidden layer's output from CBOW as a good representation of the micro behaviors.

With the embedded data, we cluster micro behaviors via the spectral clustering algorithmVon Luxburg (2007). Specifically, we will first calculate the similarity matrix $\mathbf{S}$ between micro behaviors using their embedded vectors, $\{z_i\}_{i=1}^n$. Then, we compute the normalized graph Laplacian L and compute the first k eigenvectors $u_1, \ldots, u_k$ of L. Let U be the matrix concatenating the column vectors $u_1, \ldots, u_k$, where v_i corresponds to the ith row of U. Then the cluster labels for the i-th micro behavior is achieved via k-mean clustering on the point $\{v_i\}_{i=1}^n$.

3.3.1.3 Word2vec Models for Browsing Behaviors

New technologies have reshaped people's shopping behaviors, and new buyers prefer to collect car information online, including visiting OEM websites such as Ford.com. This enables the company to collect enormous data about prospective buyers. Figure 3.2 illustrates a real example of a visitor's browsing history on Ford.com. The visitor first visited the homepage of Ford SUV exploring the inventory, price and gallery. Then, the customer moved to a sport utility car model and checked the dealer information. After that, the customer browsed the page for a mid-sized car exploring its accessories and finally went back to the homepage of Ford SUV and ended the session.

Despite its usefulness, the clickstream data's summary statistics can only capture a small fraction of the information from the clickstream data and is sensitive to other environmental factors such as launches of new models and incentives. To extract informative and robust features from clickstream data, we propose a novel word2vec as follows.

We organize the clickstream data in terms of a series of macro interactions (for different car models) and micro behavior tuples as defined above. The macro behaviors record the interactions between visitors and the models and measure the similarity between models.

For example, a customer wants to buy a family car. Even though there are multiple choices, including Sedan, SUV or Van, the customer may only compare two similar SUV models, which indicates these two models are more similar compared other car models. From a micro perspective, each macro interaction consists of a sequence of behaviors indicating what information the customer collects for the model, how long the customer dwells on a page, and whether the customer checks the information for local dealers. These micro behaviors provide additional information about the visitor. For example, the longer dwell time on a model indicates a stronger desire for the product, checking the local inventory suggests a stronger intent, and visiting model details pages, including accessory and gallery, also indicates a higher interest. One major advantage of formulating the clickstream data into sequences of micro behaviors is that it provides a framework to capture the relationship between car models and represent browsing behaviors using numerical vectors, which can be incorporated into our demand forecasting model. However, this formulation also poses tremendous challenges on 1) how to utilize the sequential nature of the data and (2) how to incorporate the click information into our forecasting model. The solution to these two challenges leads to our novel demand forecasting model.

Specifically, after organizing the clickstream data in this structured way above, we turn each browsing history into a 'sentence' with micro behaviors as 'words'. To capture the relationships between micro behaviors and embed them into a Euclidean space, we apply the word2vec framework using the Continuous Bag of Words (CBOW) model illustrated in Figure 3.1. Using CBOW on 880 days of clickstream data across the US, we learned the 20-dimensional numerical representation of each micro behavior and clustered them into 27 different groups using spectral clustering. Figure 3.5 shows the two-dimensional Uniform Manifold Approximation and Projection (UMAP) representation of different micro

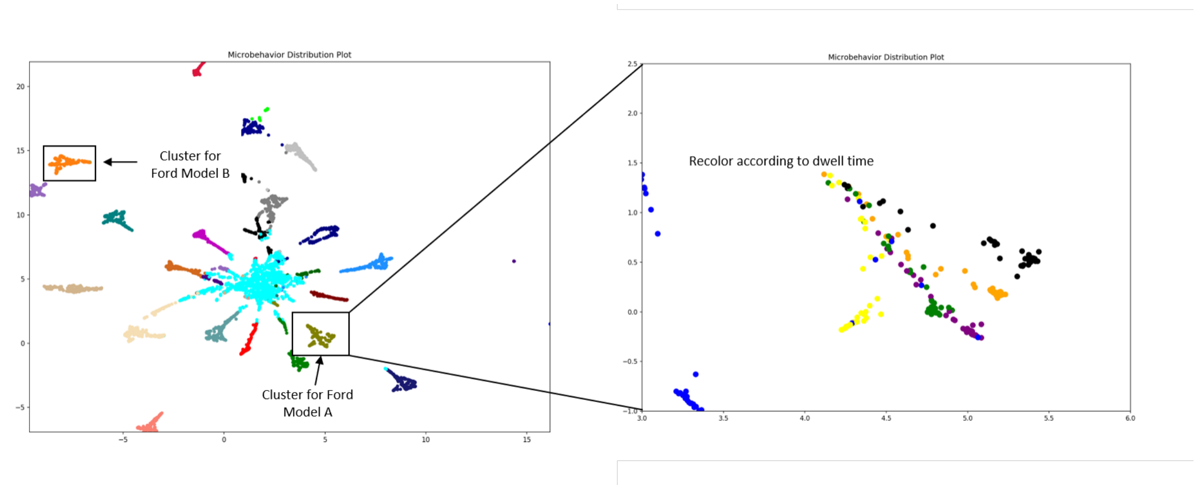


Figure 3.5: UMAP projection plot showing 27 major clusters of the 5022 micro behaviors using embedding learnt from word2vec model. The colors represents the clusters generated from spectral clustering to group them into 27 clusters.

behaviors highlighting the data's clustering structure. The microbehavior embeddings will keep the properties about car models and dwell time. In Figure 3.5, each cluster only contains microbehaviors of 1 $\sim$ 2 car models. On the other hand, microbehaviors about a certain car model can only be projected to a few certain clusters. Take Ford Model A for example, more than 95% microbehaviors about Ford Model A can only be classified into 2 clusters. It should be noticed that in the middle of the plot there exists a cyan cluster. This cluster is a chaos cluster, which represents that the microbehavior is hard to be classified. This cluster consists of rare microbehaviors and therefore it's difficult to build connection with other microbehaviors. Furthermore, we can also find dwell time properties in a certain cluster. Taking Ford Model A for example, Figure 3.5 recolored the embedding according to their dwell time category and the cluster can be further split into 5 subclusters. Therefore dwell time also player a minor role in microbehavior embedding. Besides, the microbehavior embedding also provides indication for different car models. If two cluster are near with each other, then car models in the two cluster will be considered to have some relationship with each other in respect of customers.

With this cluster information, we can extract daily robust features such as the number of incidences in cluster 2 for each zip, which can be incorporated into our CDF model.

3.3.2 Implemented Models

In our research, we improve models progressively. Firstly we select the classic statistical model Seasonal Autoregressive Integrated Moving Average as our benchmark. Then for comparison, we propose multivariate LSTM to capture the compound temporal properties in sales data. In the end, we introduce our proposed approaches to fuse the clickstream information with sales features.

3.3.2.1 Seasonal Autoregressive Integrated Moving Average Model

As a benchmark, we select the Seasonal Autoregressive Integrated Moving Average (SARIMA) model. Autoregressive Integrated Moving Average(ARIMA) is a classic way for time-series data forecasting. As there is a strong interaction between weekly effects and monthly effects, adding seasonal features can help ARIMA capture the complicated feature interactions.

3.3.2.2 Multivariate LSTM Model

To utilize the complicated temporal dependency among data, we propose a general Clickstream based Demand Forecasting Framework (CDF), which incorporates clickstream and historical sales data into a recurrent neural network model. Figure 3.6 illustrates the architecture of the proposed Clickstream based Demand Forecasting Framework.

To capture the strong temporal effects, including weekly, monthly, and holiday effects,

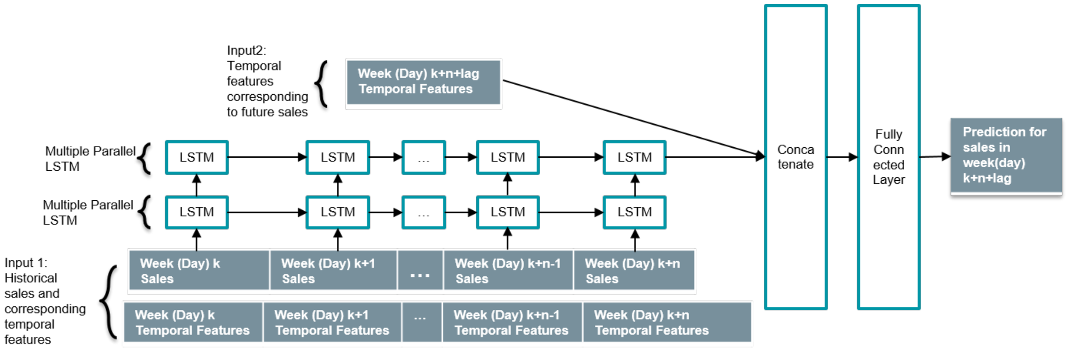


Figure 3.6: The architecture of the CDF framework.

we propose a novel multivariate LSTM based daily demand forecasting model incorporating those temporal features. Specifically, for each day, we include the following features

- the weekday information (Monday, Tuesday, etc.) using one-hot encoding
- dummy variable for whether it is the end of a month
- dummy variable for whether it is a holiday, e.g., Christmas, New year's eve, etc.

In sum, the proposed model's input features are sales and the temporal features of the past 92 days, and the output includes the sales after 30 days, as shown in Figure 3.6.

3.3.2.3 Clickstream Informed LSTM Model

Multivariate LSTM architecture provides us with the potential to tranfer information across similar domains. However, how to choose the transfer learning direction remains a challenge. Here, we propose that the interaction for online clickstream history provides an indication between car models. The interaction can be captured by analyzing the distance between microbehaviors or learned by a graph neural network. Figure 3.7 shows the graph attention network(GAT) based architecture by taking the vectored microbehaviors as a graph and multiple car models as input to improve the learning ability for single model prediction. The framework contains two modules:

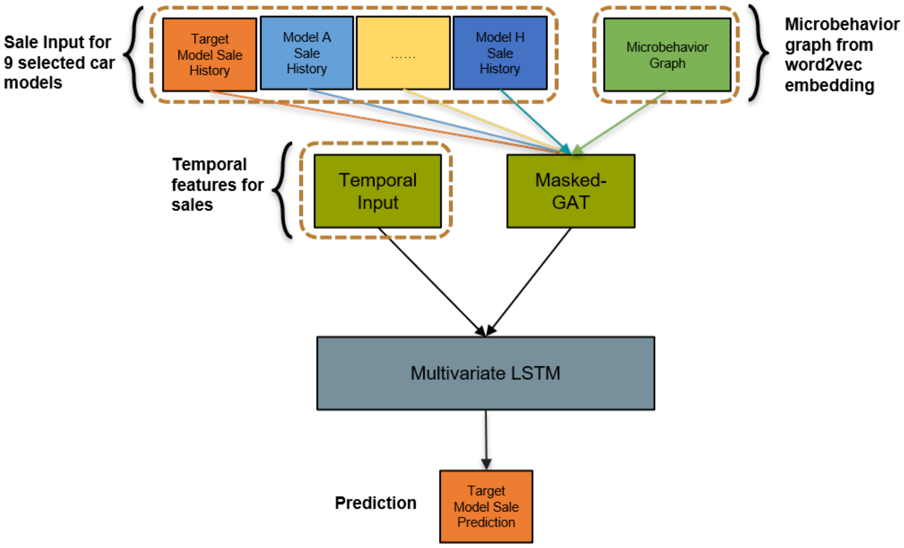

Sale Input for 9 selected car models
Target Model Sale History
Model A Sale History
......
Model H Sale History
Microbehavior Graph
Microbehavior graph from word2vec embedding
Temporal features for sales
Temporal Input
Masked-GAT
Multivariate LSTM
Prediction
Target Model Sale Prediction

Figure 3.7: The architecture of GAT-LSTM model.

- A masked-graph attention network for selected 9 car models.
- Multivariate LSTM as illustrated in Section 3.3.2.3.

Figure 3.8 illustrates the GAT framework. It will learn an attention weight to each car cluster and combine it with the cluster distance. Then according to the weighted sale history, it will generate a sequence for prediction. The multivariate LSTM part will fit the output sequence from the GAT module to predict future sales.

3.4 Experiments and Analysis

Our experiments will follow the order as our model improvement process. In Section 3.4.1 to Section 3.4.3, we first summarize the SARIMA and multivariate LSTM experiments using sales data only. Meanwhile, there also exists potential of transfer learning on Multivariate LSTM. Then we feed the summary clickstream statistics into the multivariate models directly and illustrate the performance of those different clickstream informed multivariate LSTM

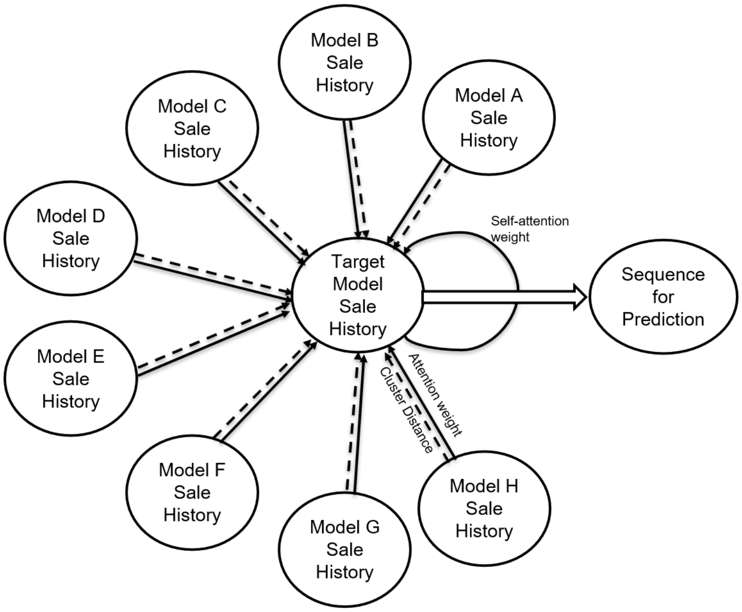


Figure 3.8: The framework of GAT module.

models in Section 3.4.4. The experiments to combine clickstream data and transfer learning are covered in Section 3.4.5.

3.4.1 Benchmark Comparison

First we compared the performance of SARIMA model and multivariate LSTM model in car sales forecasting. Then we run transfer learning experiments on different levels with the multivariate LSTM framework.

Seasonal Autoregressive Integrated Moving Average Model Settings

To decide the hyperparameters within the SARIMA model, we apply the autocorrelation function and find where there exists a high correlation every 7 days. Within each week, there is no significant seasonality. Therefore we decide to set $p = 7$. From the PACF plot, we can see that there exists a lag of 7 days, and there is less correlation when the lag is higher than 7. This indicates a selection of q as 7. To address if we need to apply differencing within the series, we tried several settings for d to perform differencing for historical sales and checked its corresponding Akaike information criterion(AIC). Within all settings, $d = 0$ had the lowest AIC. Therefore, we choose the hyperparameters for the ARIMA model as $p = 7$, $d = 0$, $q = 7$. Meanwhile, to capture the monthly effects, we also added the seasonal parameter as 30 days in SARIMA.

Multivariate LSTM Model Settings

We implement the Asynchronous Successive Halving Algorithm (ASHA), a simple and robust algorithm for hyper-parameters tuning in multivariate LSTM model and GAT-LSTM model. The 2-layer LSTM contains 5 and 3 kernels, and the fully connected layer after the concatenate layer has 5 nodes. The training algorithm is adam with the learning rate as 0.0001.

3.4.2 Daily Demand Forecasting

We first test the performance of our model on the national level data. We split the national sales data into a training set (09/07/2016-01/20/2018), validation set (01/21/2018 - 04/30/2018), and testing set (05/01/2018 - 10/11/2018).

Traditionally, people tend to use Mean absolute percentage error (MAPE) to measure the model performance. The MAPE is defined as

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\frac{|\hat{y}_i - y_i|}{y_i},$$

where y_i is the true sale on ith day, and $\hat{y}_i$ is the predicted sale. However, in our sale data, there are many days with sales close to zero as shown in Figure 3.3, which makes MAPE metric statistically unstable. Instead, we introduce a new metric, termed L^2-MAPE, to measure the model performances, where

$$L^2\text{-MAPE} = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{n} y_i^2}}.$$

3.4.3 Transfer Learning on Multiple Levels

To improve the model and leverage the information from other regions, we propose to incorporate the transfer learning framework into our framework. Here, transfer learning is a system's ability to recognize and apply knowledge and skills learned in previous tasks to novel tasks or new domains, which share some commonality. The following experiments summarize the transfer learning potential for the multivariate LSTM model.

Region-level Transfer Learning

In our setting, the sales data in regions with a similar culture and incentive schedules tend to inherit similar patterns, which can improve the model accuracy for a specific region. To implement the transfer learning framework, We first train our model on national data and record the learned weights/parameters. We then initialize our local model using the national data's weights and train the model with data from each state. The implementation of the transfer learning leads to reduction in most states with respect to L^2-MAPE loss.

Figure 3.9 compares L^2-MAPE of SARIMA, randomly initialized LSTM and LSTM initialized from the trained national model in all 54 states and territories. The X-axis represents all states ordered by their regional car sales. In most states, LSTM has higher accuracy than SARIMA, and initializing weights from trained LSTM model can further improve model performance.

Model-level Transfer Learning

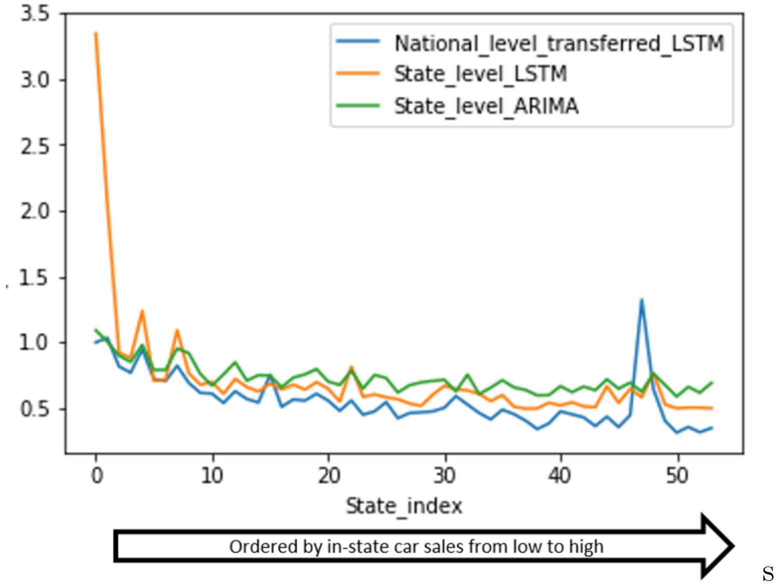


Figure 3.9: L^2-MAPE Comparison in car model B State Level Prediction.

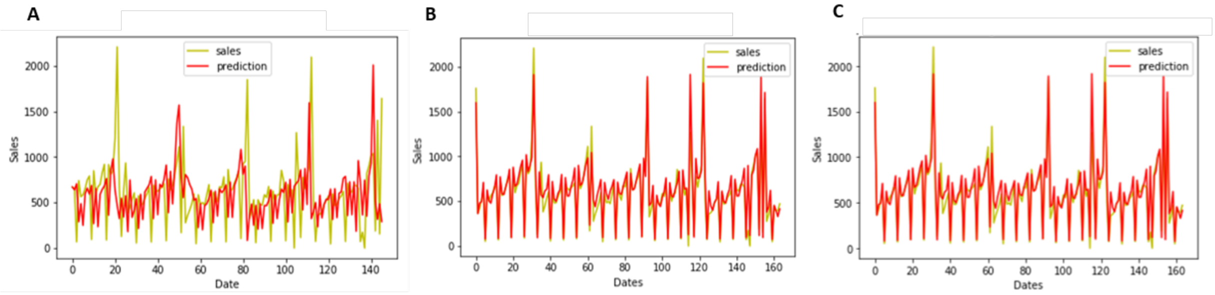


Figure 3.10: Daily demand forecasting for car model B in the US. A: Forecasting performance of SARIMA model. B: Forecasting performance of Multivariate LSTM model. C: Forecasting performance of Multivariate LSTM model with weights transferred from car model A.

Models	**L2-MAPE**
SARIMA	57.10%
Randomly Initialized Multivariate LSTM	19.45%
Multivariate LSTM Transferred from car model A	**19.00%**

Table 3.1: Daily demand forecasting accuracy of model B.

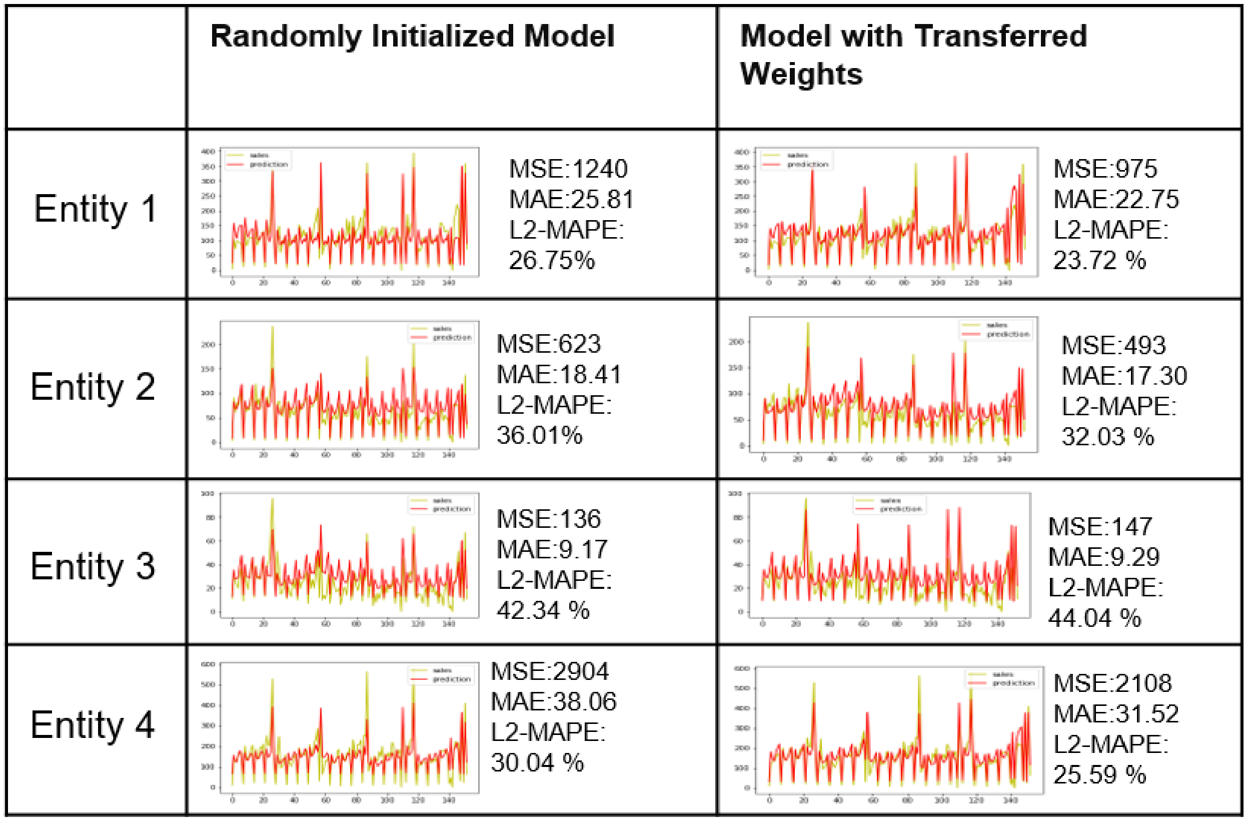
Models	**L2-MAPE**
SARIMA	48.83%
Randomly Initialized Multivariate LSTM	23.69%
Multivariate LSTM Transferred from model B(less popular)	23.65%
Multivariate LSTM Transferred from car model A(more popular)	**19.14%**

Table 3.2: Daily demand forecasting accuracy of car Model C.

Besides transferring information within the same model, an interesting question is whether we can transfer information between models. For example, can we transfer information from the best selling model, to our target one or multiple models? We pick up car model A, B and C from Ford popular compact car models, where A sells much more than the other two car models. Figure 3.10 and Tables 3.1, 3.2 demonstrate that the transfer information from a more characteristic domain can improve the Multivariate LSTM model prediction accuracy for both car model B and C. In addition, transfer learning from different domains shows various performances. Table 3.2 shows that the weight initialization from the trained model for car model A and B can help improve the prediction accuracy of C.

Entity-level Transfer Learning

There are multiple entity levels within the same car model. In Ford dataset, entity information is matched with a unique hashed VIN number for some car models. Then, by matching the hashed VIN number with the historical car sales, we grouped one kind of Ford SUV model sale history by entities. It should be noticed that after grouping, the distribution of some entities will be very sparse with respect to time. Some entities are directly related to model years, so only the time around the model year will have these kinds of entities recorded. We picked up 5 entities recorded in at least 3 years and compared the model performance between the random initialized multivariate LSTM model and the multivariate


sales
prediction

	Randomly Initialized Model	Model with Transferred Weights
Entity 1	MSE:1240 MAE:25.81 L2-MAPE: 26.75%	MSE:975 MAE:22.75 L2-MAPE: 23.72 %
Entity 2	MSE:623 MAE:18.41 L2-MAPE: 36.01%	MSE:493 MAE:17.30 L2-MAPE: 32.03 %
Entity 3	MSE:136 MAE:9.17 L2-MAPE: 42.34 %	MSE:147 MAE:9.29 L2-MAPE: 44.04 %
Entity 4	MSE:2904 MAE:38.06 L2-MAPE: 30.04 %	MSE:2108 MAE:31.52 L2-MAPE: 25.59 %

Figure 3.11: Entity-level model comparison.

LSTM model with the weight initialized by the trained model for all entities. Figure 3.11 shows that out of 5 entities, 4 of them have a better performance after transfer learning.

3.4.4 Clickstream Informed Multivariate LSTM Model

We cleaned the clickstream data and fed the data into our multiple-input LSTM model. We evaluate the model performance on selected Ford SUV in 2017∼2020 and summarize the results in Table 3.3. The following shows the input settings for the merging degree of clickstream statistics and model. The results are shown in Table 3.3

- **Baseline** Inputs only includes the car sales and temporal property features.

- **Model 1:** Naive Clickstream Input Model. We construct the clickstream input by feeding

	MSE	**MAE**	**L2-MAPE**
Baseline	28908	120	24.38%
Model 1	27303	111	23.68%
Model 2	**26802**	120	**23.46%**
Model 3	27624	**108**	23.82%
Model 4	39039	154	28.34%

Table 3.3: Performance for clickstream informed multivariate LSTM models.

all microbehavior tuple into the Word2vec model and do clustering. Then we pick up the clusters according to the car model in microbehavior tuples. We calculated the number of microbehaviors such clusters and fed it into the model along with the sales.

- **Model 2:** Cleaned Naive Clickstream Input Model. There exist uninterpretable and synonymous records in the clickstream dataset. We manually removed or merged those records before feeding them into the model.

- **Model 3:** Concise Clickstream Input Model. To make a better use of dwell time information, we picked up the core 13 actions and calculated the summation of the dwell time for each action. We also set a threshold for the dwell time to be no more than 120 seconds.

- **Model 4:** Concise Clickstream Input on R-Transformer. We updated the RNN part of the model from LSTM to R-transformer as illustrated in the state of the art sequential model Wang *et al.* (2019d). The input is the same as the cleaned naive clickstream input model.

As a result, after proper cleaning, adding clickstream summary statistics into the multivariate LSTM model can improve the model performance.

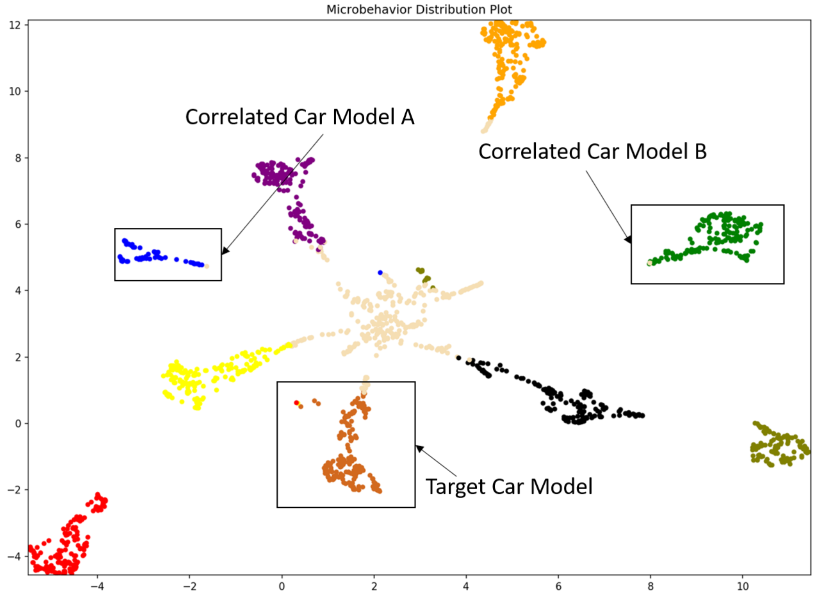


Figure 3.12: UMAP projection plot showing 10 major clusters using embedding learned from word2vec model. The colors represent the clusters generated from spectral clustering to group them into 10 clusters. The middle wheat-colored cluster represents the chaos part. Other surrounding clusters each represent a certain car model.

3.4.5 GAT-LSTM Model

3.4.5.1 Experimental Results

We take a less popular car model as our target and select 9 popular car models as our graphic input. The distance between models is derived from the distance between the microbehavior cluster's midpoints corresponding to each model in the Word2Vec result. The sale history ranges from Oct. 27, 2016 to Nov. 17, 2020. As the GAT model contains more parameters than the LSTM model(4196 vs. 474), constructing samples from only 1482 days can't guarantee enough samples. Therefore, we also manually select 20 states with high sales and run experiments on those states.

To avoid the effect of local optimum during a single training process, we switched random seeds and run 50 experiments. Table 3.4 summarized the average results. GAT-LSTM outperforms Multivariate LSTM model for the 20 selected states and the summation of the

L^2-**MAPE**	**State Level Prediction**	**Summed Prediction for all 20 States**
Multivariate LSTM	64.04%	7.84%
GAT-LSTM Model	**61.73%**	**7.57%**

Table 3.4: Results for state-level car sale prediction. We ran 50 times under the same settings and calculated the average L^2-MAPE.

car sales in these states.

3.4.5.2 Graphical Analysis

Graphic results from the clickstream data also provide us with a straightforward direction of transfer learning. Figure 3.12 shows the clustered Word2Vec embedded result, the targeted model and correlated car model A shows the highest correlation, while car model B shows the almost lowest correlation with the target car model. It should be noticed that the UMAP plot doesn't necessarily represent the exact distance between different car models. According to the Word2Vec embedding results, car model A is the nearest cluster to the target car model, while car model B is the furthest cluster. We compared the multivariate LSTM model performance by initializing weights from the trained model of car model A and the trained model of car model B. Table 3.5 shows the performance's summary under different weight initialization. Weight initialization from car model B shows a worse performance even than random initialization. This indicates the trained car model B model leads to an opposite direction for the target car model forecast. Meanwhile, car model A initialized model performs better than the randomly initialized model. Therefore, the direction of the clickstream data provides a more reliable way for transfer learning.

Weight initialization method	**L^2-MAPE**
Random initialization	24.17%
Weight initialization from car model B	27.61%
Weight initialization from car model A	**23.12%**

Table 3.5: Performance comparison of multivariate LSTM model. Weight initialization are derived from random generation, trained model of car model B and trained model of car model A.

3.5 Discussion

We have 1) built an LSTM based demand forecasting model with the capacity to provide accurate daily sale prediction on both national and local levels for each model; 2) implemented the word2vec model to extract customer micro behavior feature from clickstream data; 3) Proposed a graph neural network method to capture the relationship across different car models. Applying the clickstream data for future car sales is still at an early stage. There are two possible directions to improve the performance of this network. 1) Noise reduction for clickstream data and historical sale limited sample size. 2) Methodology to combine the clickstream data and sale history.

Chapter 4

Machine learning towards multimedia data

In this chapter, we will talk about the understanding of the data in multimedia field. One important application of the multimedia data is advertising. With the development of the multimedia technology, rich media advertisement is more and more popular in business. Since the rich media services needs the guarantee the fitness across different modalities, we would like to build a model to analyze the correspondence between different modalities. In this paper, we mainly includes 2 topics:

- How to learn the correspondence across multiple modalities?
- How to arrange the information flow within the model to achieve the best performance?

To answer the first question, we introduced themes, an outside modality that helps to project the embedding vectors for different modalities into the same projection space. Then based on the new model, we propose a self-adapted training algorithm to let the model optimize its architecture with limited parameter size.

4.1 Themes informed audio-visual correspondence learning

Recently, the applications of short-term user-generated video (UGV) boom quickly, such as Tik Tok, Youtube short-term videos, Snapchat and Kwai. In multimodal field Baltrušaitis *et al.* (2018), an important application is audio-visual correspondence (AVC) learning, which can tell whether or how well the audio and visual information matches in the video. It can recommend audio or visual streams to users given the other modality that contributes to the same target, evaluate the quality of the short-term video for pushing to users, and build a better high-level representation of videos for other uses.

Efforts have been made on the AVC learning. A general idea is finding a shared projective space for multiple modalities. Given an modal embedding vector space V, and let $\mathbf{P}(V)$ be a projective space through a canonical map $p : V \rightarrow \mathbf{P}(V)$. We need to find a proper function f, and take

$$Corr = f(\mathbf{P}(V_{audio}), \mathbf{P}(V_{visual}))$$

as an estimation of the correspondence of audio and visual information. However, most previous works have limitations, mainly lying on two shortages: the task setting was simple, such as matching background audio to single image Aytar *et al.* (2016); and the approaches relied on simple assumptions that audio and visual information should be similar in the projective space Arandjelovic and Zisserman (2017); Li and Kumar (2019); Zhu *et al.* (2021b). These shortcomings may fail the systems on UGVs, where the video can convey more than one themes by switching modality combinations, and the confusion between those combinations may introduce too much variance into the model. Therefore, in complex AVC learning problems, we should also decide whether the correspondence measurement is proper with respect to a certain theme. One example can illustrate this: the user may combine a series of cheerful wedding ceremony photos and low-spirited style music to present the theme “marriage is the tomb of love” – spoken by Giacomo Casanova.

To model the complex relationship between visual and audio information, we introduce the concept of themes and propose a Theme Informed AVC (Ti-AVC) learning algorithm. Ti-AVC involves themes as an important auxiliary modality to learn the projective space as following:

$$Corr = f(\mathbf{P}_{theme}(V_{audio}), \mathbf{P}_{theme}(V_{visual}))$$

To establish the themes-informed projective space, we propose that the matched audio and visual information should follow two principles:

1. both modalities should convey the same desired theme;

2. there exists positive interactions between them when presenting the theme.

For the first principle, we designed a a novel framework to inject the theme information into AVC learning. Since it is not clear how to represent the theme, we adopted the video tags to direct model the theme indirectly in this project. For the second principle, we followed conventional ideas and adopted a state-of-the-art framework to model the relationship.

To evaluate our proposed framework, we collected 85432 UGVs from Kwai, a popular short-term video app in China. All the collected videos are advertisement (ads) uploaded by commercial advertisers. We will publish the dataset as the extension of KWAI-AD Chen *et al.* (2020) dataset. In this dataset, our proposed approach gained 23.15% improvement in accuracy AUC compared to a state-of-the-art AVC learning framework.

We summarize the contribution of this project below: 1) We introduced new principles for the AVC learning task. 2) We proposed the first theme informed audio-visual correspondence (Ti-AVC) framework which is suitable for UGVs. It outperformed the state-of-the-art baseline by 23.15% absolute difference, and its hidden values indicate the modality information flow in AVC. 3) We published the first audio-visual dataset grouped by contents based on short-term ads video.

4.1.1 Related Work

Researchers cast much attention on reciprocity between audio and visual information on various tasks Tao and Busso (2020). Although transfer learning has been proposed to convey information across modalities, how to model the correspondence between modalities is still an open question.

L^3 net Arandjelovic and Zisserman (2017) was proposed to explicitly model AVC. It used several sub-networks to perform inputs processing and modalities fusion. Relying on the max-pooling layer in the fusion sub-network, the L^3 net had a flexible framework that was able to take sequential or single input. It showed state-of-the-art performance and a new perspective to perform sound localization taskArandjelovic and Zisserman (2018); Wu *et al.* (2019b). In audio-visual cross-modal embedding designs, the pre-trained L^3 net is deployed as embedding extractor Cramer *et al.* (2019); Chung *et al.* (2019). Verma et al. applied L^3 net framework to learn AVC based on the emotion from audio and visual streamsVerma *et al.* (2019). This work was evaluated on a new released dataset that contained audio and visual emotion information. To increase the correspondence, dual attention matching Wu *et al.* (2019b) added attention to both audio and visual inputs to predict their event sequential localization relevance between modals; elastic multi-way network Wang *et al.* (2019b) designed a loss function with the distance between samples and an anchor point to encourage correspondence; Tao and Busso (2018b) relied on a bimodal recurrent neural network to learn the temporal correspondence information in a data-driven fashion. Unsupervised methods such as video audio correspondence were also investigated such as audio-visual deep clustering modelLu *et al.* (2019). Most of the approaches focused on modeling the similarity between modalities and showed decent performance. However, most of the approaches were only evaluated on constrained dataset. AVC learning on unconstrained data is still a complicated and difficult task. Zhu *et al.* (2021a); Baltrušaitis *et al.* (2018).

There are several public available unconstrained audio-visual datasets such as UGVs datasets, but none of them are suitable for the short-term videos case. Specifically, Youtube-8M Abu-El-Haija *et al.* (2016) only, one of the most popular UGVs dataset, covers various themes (i.e. tags), but its video quality is not controlled intentionally. Also, the video duration in Youtube-8M exceeds the typical length for a short-term video. On the other side, the Moments in Time Monfort *et al.* (2019) contains 1,000,000 3-second videos, which is too short. Flickr-SoundNetAytar *et al.* (2016) is a unconstrained dataset, however it only has single image with background sound track. Movienet and HVUDiba *et al.* (2020); Huang *et al.* (2020) introduce some holistic datasets, but their audio-visual properties are not significant. The shortage of good quality short-term UGVs inspired us to collect a new data, whose details will be introduced later.

4.1.2 KWAI-AD-AudVis Dataset

In this study, we developed our framework on KWAI-AD-AudVis dataset. It constists of 85432 ads videos (around 913 hours) from the China popular short-term video app, Kwai. The videos were made and uploaded by commercial advertisers. The reason to use the ads videos lied on two folds: 1) the source guarantees videos under control to some level, such as high-resolution pictures and intentionally designed scene; 2) ad videos simulate audio-visual matching style as manually composited by users in Kwai app. It can be seen as a quality controlled UGVs dataset.

In the KWAI-AD-AudVis dataset, each UGV/ad has a label for the industry category. We estimate the number of clicks advertisers receive every time the ads come out as a criteria during collection. In this dataset, half of the ads have a high rate to raise customers' interests in the products, and the other half has a relatively low attraction. The short videos have been classified into 19 themes by uploaders with an average length of seconds. The audio track had 2 channels (we mixed to mono channel in the study) and was sampled at 44.1 kHz, while the visual track had a resolution of 720 × 1280 and was sampled at 25 frames per second (FPS). This dataset is an extension of the KWAI-AD corpus Chen *et al.* (2020). It is not only suitable for tasks in the multimodal learning area, but also for ones in ads recommendation.

The details and data of KWAI-AD-AudVid can be accessed through Zenodo[1]. It shows that the ads videos have three main characteristics: 1) The videos may have very inconsistent information in visual or audio streams. For example, the video may play a drama-like story at first, and then present the product introduction, whose scenes are very different. 2) The correspondence between audio and visual streams is not clear. For instance, similar visual objects (e.g. talking salesman) come with very different audio streams. 3) The relationship between audio and video varies in different industries. For example, games or E-commerce ads will have very different styles. These characteristics make the dataset suitable yet challenging for our study on AVC learning.

[1] https://zenodo.org/record/4023390#.X12Dr5NKgUE

4.1.3 Proposed Approaches

Data and Feature

In this study, we used KWAI-AD-AudVis dataset to develop our AVC learning framework. To reduce the training workload, we used our in-house key-frame extractor to extract 8 frames from each video to represent the visual information. Audio tracks were extracted as same as in original videos. The visual and audio information are pre-processed through Mobilenetv2 Sandler *et al.* (2018) and VGGish Hershey *et al.* (2017). Embedding from top layers of the two pre-trained was fed to our proposed system.

Themes Informed AVC learning System

Figure 4.1 shows the diagram of our proposed approach, theme-informed audio-visual correspondence (Ti-AVC) learning framework. It consisted of two parts, a theme-learning (TL) model and a correspondence-learning CL model.

For the TL model, we were inspired by L^3 net and designed a similar network as L^3 net, except its task was theme prediction (in this study, it is ads industry category prediction). It took audio and visual embedding as input. It consisted of three sub-networks. Two sub-network processed the input of single modality separately, and the third one processed the fused information. The audio sub-network had a time distributed dense layer, an LSTM layer and a self-attention layer. Its output is a 128-D vector. The visual sub-network had a fully connected layer, whose parameters were shared across different input frames. Its output was a sequence of 8 128-D vectors. The output from the audio sub-network was repeated and concatenated to each vector from the visual sub-network. The concatenated embedding was fed into the fusion sub-network. The fusion sub-network had two 1-D convolutional neural networks (CNN), a max-pooling layer and 2 fully connected (FC) layers to predict themes.

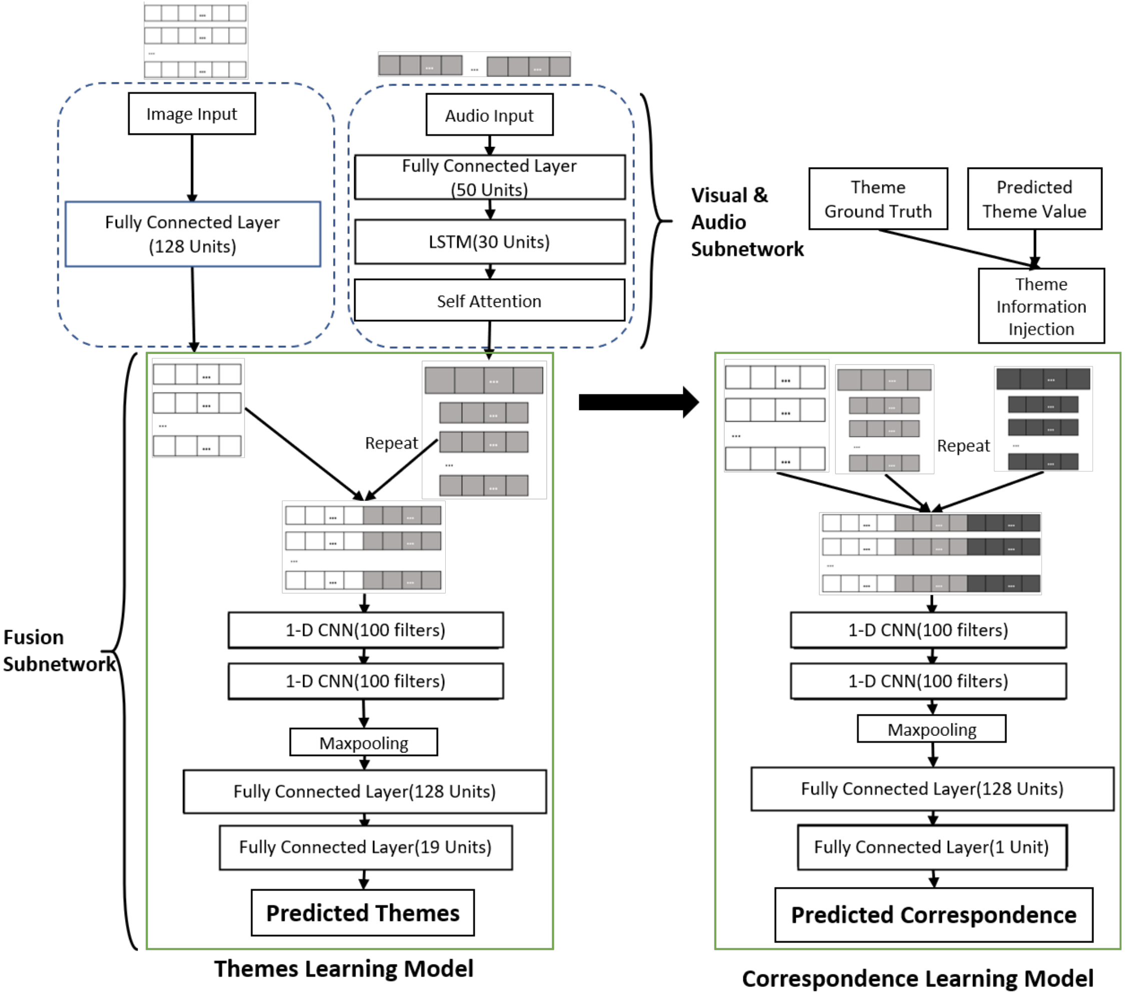


Figure 4.1: Diagram of the proposed framework.

Once the TL model was trained, we fixed it as an embedding extractor to extract three types of information for the CL model: audio embedding from the top-layer of the audio sub-network, visual embedding from the top-layer of the visual sub-network and the predicted theme. We concatenated the predicted theme with the theme ground-truth to form the theme information, which was injected into the CL model with audio and visual embedding. By adding the true theme, we expect the CL model to learn how the input audio and visual embedding performs in predicting the theme. This was following the principle (1) we mentioned in the introduction section. In this study, the theme ground-truth corresponded to the visual modality. The CL model has similar architecture as the fusion sub-network in the TL model. We intended to use both of the theme prediction and ground-truth to tell how the two modalities represent the desired theme. The CL model was expected to capture two points: 1) how the desired theme was presented; 2) how the modalities related to each other. These two points corresponded to the two principles we proposed at the beginning of this section. The correspondence result was eventually predicted based on the two points.

4.1.4 Experiment and Analysis

Experiment Setup

We used the original videos from the KWAI-AD-AudVis dataset as positive samples, where we assumed audio and visual information matched with each other. Negative samples were generated by pairing audio and visual tracks from different videos. We generated the same number of negative samples as positive ones. The dataset was partitioned to 80%, 10% and 10% for training, validation and testing respectively. We applied Adam as the optimizer and set the learning rate as 0.0001, batch size as 8 in all experiments. We used ads industries categories as theme information in this study.

We built two baselines for comparison. The first baseline (denoted as “baseline-1") borrows the architecture from the themes learning model. To make a fair comparison, we made two adjustments for correspondence learning: 1) we replaced the theme prediction task by correspondence prediction; 2) we doubled the number of all trainable layers in the fusion sub-network to guarantee the same parameter size as our proposed approach. The second baseline (denoted as “baseline-2") had exactly the same architecture as baseline-1, except we input theme ground-truth to the fusion sub-network concatenated with the modalities embedding. This made the comparison fair since the system also got theme information like the proposed approach. For the proposed approach, we made two training strategies and therefore had two systems. We named the system that trained TL and CL models separately as “Ti-AVC", while we named the one jointly trained (i.e. a multitask learning system with TL and CL tasks) as“joint Ti-AVC". We kept all systems having the same number of parameters.

Experiment Results

The accuracy AUC score of each system is shown in Table 4.1. The baseline-1, which had similar architecture to L^3 net, had random-guess results (we had the same amount of positive and negative samples). The baseline-2, which was the same as baseline-1 except it took the theme as an extra input, could outperform baseline-1 by 18.94%. This verified our hypothesis that theme information was necessary for AVC learning on UGVs. Both of our proposed approaches beat the baselines (by at least 3.36% absolute difference) with the Ti-AVC achieving the best performance. Since the TL and CL models were trained separately in Ti-AVC, it indicates that properly injecting the information on how the audio and visual modalities presented the desired theme could improve the performance of the correspondence learning task. We would like to emphasize that the Ti-AVC is flexible in application. The TL

Model	Match AUC
Baseline-1	55.58%
Baseline-2	74.52%
Joint Ti-AVC	77.88%
Ti-AVC	**78.73%**

Table 4.1: Summary of experiment results.

model can be fixed as embedding extractor and the theme categories provided by CL model can be obtained from either modality (in this study, we made it follow visual modality).

We also performed evaluations within each theme category (shown in Figure 4.2), where all the testing candidates were from the same category in AUC computation. Since all the testing candidates had the same theme ground-truth, the evaluation was equivalent to eliminating the information of theme ground-truth. The CL model can only obtain help from the difference between the theme prediction and ground-truth. This difference can represent "how the desired themes are presented" as we proposed in principle (1), so the results can reflect the effectiveness of the principle (1) in the AVC task. We compare the results with the baseline-1 (the horizontal line in Figure 4.2), which did not include theme ground-truth during inference. The result shows that the Ti-AVC framework dominates the baseline in 15 categories out of 19. Especially, we notice all the categories with most samples outperformed the baseline. This result indicates that the proposed framework can help improve the correspondence learning even without theme information, and justify the proposed principle (1).

Contribution Analysis

To further verify the rationality of our proposed approach, we analyzed the contribution of each input in the CL model. We use the information flow fed into the convolutional layers to indicate the importance of each modal. As both positive and negative values have an

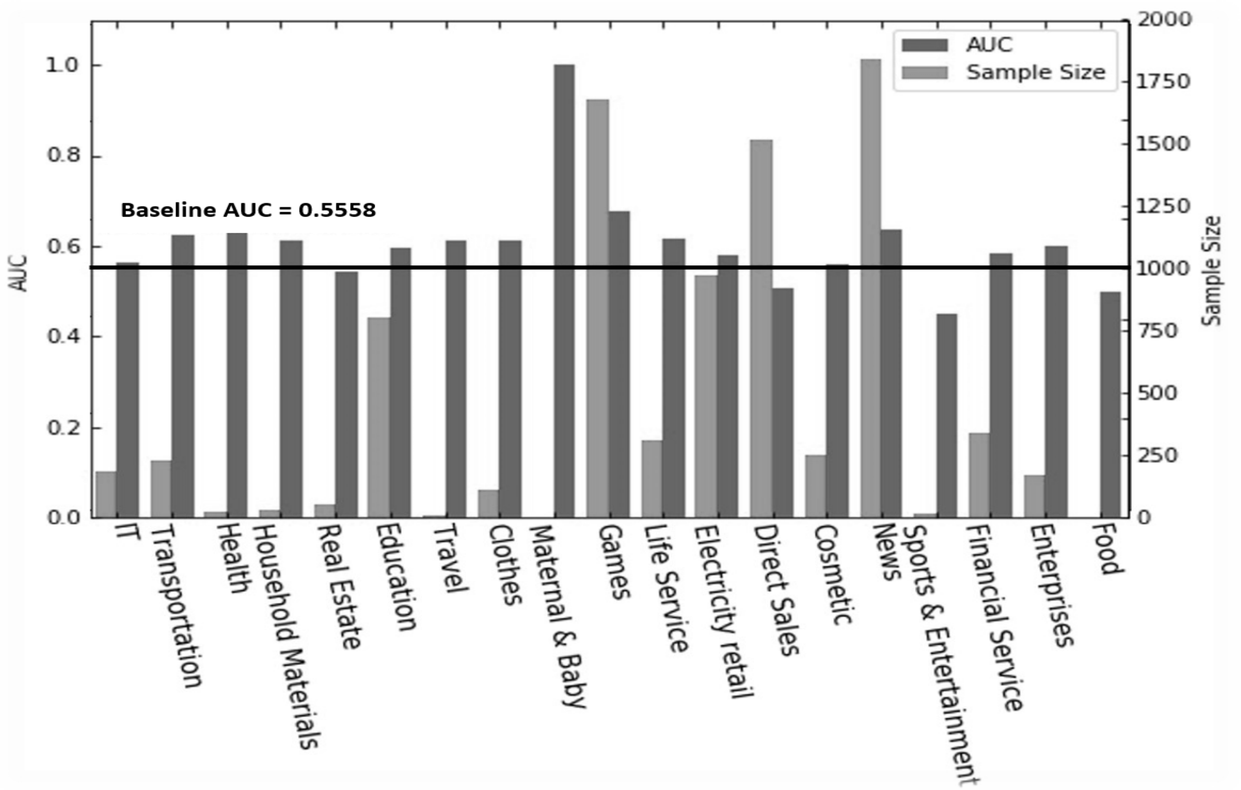


Figure 4.2: AUC and sample counts per ADs category. The dark grey bar represents the AUC, whose scale axis is on left; the light grey bar represents the number of samples, whose scale axis is on right. The horizontal line is the baseline-1 accuracy AUC.

effective influence on the prediction, we set the absolute value of the input values as our estimation statistic. Define the contribution in equation 4.1, where W_i is the weight of the first layer connecting the i_{th} input and X_i is the i_{th} input, I is the input type (audio, visual, predicted theme and true theme).

$$Contribution^I = \sum |W_i^I \cdot X_i^I| \tag{4.1}$$

Table 4.2 listed the computed proportion of the inputs for the matched pairs. It showed that audio modalities have the most portion contribution (58.78%). The theme information took up 10.38%, where the predicted and true ones were close (4.52% and 5.86%). This result indicated both of them could not be neglected, which verified our proposed principles

Vision	Audio	Predicted Themes	True Themes
30.85%	58.78%	4.52%	5.86%

Table 4.2: Modal contributions calculated from a batch of positive audio-visual pairs and a batch of negative audio-visual pairs.

for AVC and the capability of the proposed approach.

4.1.5 Discussion

In this project, we proposed new principles in audio-visual correspondence learning on users generated videos, which introduced theme information in AVC tasks. We proposed a new framework to perform the AVC task under unconstrained scenarios. To evaluate the proposed approach, we also collected and released the KWAI-AD-AudVis corpus, consisting of 85432 short-term videos (around 913 hours).

Our proposed approach was able to outperform a state-of-the-art AVC framework by 23.15% in accuracy AUC. We also showed that the proposed approach could still outperform the baseline even without theme information. Besides, the proposed framework would be flexible in real application as the TL model can be fixed and the theme information can correspond to either modality. This study only focused on learning correspondence between audio and visual modalities by concatenating the embedding of the modalities. The future work lies on a more sophisticated fusion strategy and further analyzing how the modality correlates with each other.

4.2 Self-organized short video advertisement evaluation system

As *daily active users* (DAU) of video sharing apps, e.g, Youtube, Snapchat and Kwai, have rocketed in recent years, advertisers take advantage of this trend to promote their products or services through *user-generated videos* (UGV)-based advertisements. Generally, user behavior-related metrics, e.g., *click-through rate* (CTR)[2], 3-second play rate[3], are employed to assess advertisement quality and performance. These two metrics can be calculated as follows: *CTR = number of clicks*[4] $\div$ *impressions*[5] ; and *3-second play rate = number of plays (more than 3s)* $\div$ *impressions.* Recent researches Wang *et al.* (2019c,a) on recommender system require user profiles, which are extracted from user browsing history and user basic information, as essential model input to make precise prediction on video CTR within each user account. However, in some application scenarios, such as cold start and automatic generation of advertisements, where user-related information cannot be obtained, advertisement publishers have to rely on video content to estimate advertisement performance. Accordingly, methods for making precise prediction on UGV-based advertisement performance without user profile are of great value. *Multi-Modal Machine Learning* (MMML), which exploits signals of different modalities jointly, is able to help in mentioned video-related tasks.

Application of MMML tasks has been widely studied, including emotion recognition Tao *et al.* (2018); Liu *et al.* (2018), object localization Arandjelovic and Zisserman (2018); Zhao

[2] CTR describes how many users become interested in product based on video content.

[3] 3-second play rate describes how much the users are attracted by the content of the beginning three seconds.

[4] The event that an user click the link that comes with advertisement.

[5] Impression here refers to the count of the event that a video is fetched from dataset and recommended to users.

et al. (2018), speech recognition Tao and Busso (2018a); Afouras *et al.* (2018); Tao and Busso (2018c), speech separation Wu *et al.* (2019a), voice activity detection Tao and Busso (2019, 2020) and etc. We notice that multi-modal fusion strategy plays a decisive role in multi-modal tasks. Previous works have proposed sophisticated fusion methods and achieved remarkable success. However, in our case, previous solutions have two limitations: 1) they mainly focus on signal perception-related tasks, rather than user behaviors; 2) it is unclear how modalities interact with each other. For example, in ASR task Tao and Busso (2018a), visual content is taken as auxiliary information for audio content and therefore, audio modality is taken as query information in attention model. However, in our case, we have no prior knowledge about the relationship between input modalities and use behaviors. These shortages may lead to difficulties of applying existing methods in computational advertising.

This study focus on building an end-to-end system for predicting CTR and 3-second play rate of UGV-based advertisements. The contribution of our work can be summarized as follows: 1) To the best of our knowledge, our proposed system is the first work of predicting CTR and 3-second play rate directly from video content (combining audio and visual modalities), which is equivalent to predicting user behavior directly from raw signals. 2) We propose a self-organizing system that is able to learn the optimal topology of neural network architecture. More specifically, it is a data-driven framework which can adjust information flow by changing model architecture.

We evaluated our proposed method on a video dataset which consists of 9841 advertisements videos collected from Kwai, a trending short video app worldwide. All videos are uploaded by advertisers and contain unconstrained information. The experimental results of CTR and 3-second play rate prediction reveal that the proposed method outperforms all models for comparison.

4.2.1 Related Work

To build the first framework for predicting user behavior from videos, we borrowed ideas from recommender system and MMML studies. Recommender system relied on designated data pre-processing to collect descriptive features. Google's Wide-&-Deep model Cheng *et al.* (2016), which has been widely deployed in industry, combined these features at different levels within one neural network. Recently proposed AutoCTR framework Song *et al.* (2020) explored the optimal model structure in a data-driven way. These ideas inspired us to design a model that is able to process features collected from different levels. However, AutoCTR and Wide-&-Deep model may not be applicable in our task, as they could not handle raw signal inputs. *Densenet* Huang *et al.* (2017) essentially had similar strategy to Wide-&-Deep model that it used cross-layer connection in image classification task. It merged information from different levels in *dense block*, where one layer was directly connected to all its subsequent layers.

In MMML research, one straight-forward fusion method was to combine weighted prediction results across all modalities, where the weight of each modality was determined by its own performance on validation set. This method may fail in handling UGVs, whose modalities had different significance across UGV topics. Another widely applied fusion method was to concatenate the features extracted from different modalities into a joint representation Noroozi *et al.* (2017); Afouras *et al.* (2018); Wu *et al.* (2019a) and then the concatenated feature vectors could be processed by a classification/regression model. Such simple method has shown its effectiveness in many video-related tasks introduced in previous section. However, merging features into one vector did not provide enough flexibility in dealing with unconstrained videos. In many other works, attention model Vaswani *et al.*

(2017) has been considered Hu *et al.* (2019) to assign weights dynamically, where the strategy could be learned through end-to-end training. Also signal from one modality could be utilized as auxiliary information on other modalities Tao and Busso (2018c); Yu *et al.* (2020). These methods made prior assumptions on relationship among all available modalities and set constrains on the architectures of fusion models, which was not the case in our study. To tackle the shortages mentioned above, we develop a self-organizing framework, which is able to explore the optimal topology of neural network architecture through training data. Our proposed method bridges these two research domains so that it is able to make prediction on user behavior with original video input.

4.2.2 Dataset

In our study, we use our in-house dataset, containing advertisement play history within one week. The advertisements have been grouped into 19 pre-defined categories by advertisers. This video setting is same as KWAI-AD Chen *et al.* (2020) dataset. All audio tracks are sampled 44.1kHz sampling rate and each audio track has two channels. We mix each track into mono-channel in this study. The visual tracks have the resolution of 720 × 1280, a typical vertical setup for mobile devices. All advertisements have the same *frame per second* (FPS) of 25. Also, we summarize one-week performance for each advertisement, including impression, CTR and 3-second play rate.

However, CTR and 3-second play rate are lack of statistical significance without enough impressions. Therefor, based on our experience, we set 70,000 as impression threshold, under which advertisement samples have been discarded. After this step of filtering, a total of 9841 advertisements are collected. The total length of these advertisements is about 82 hours.

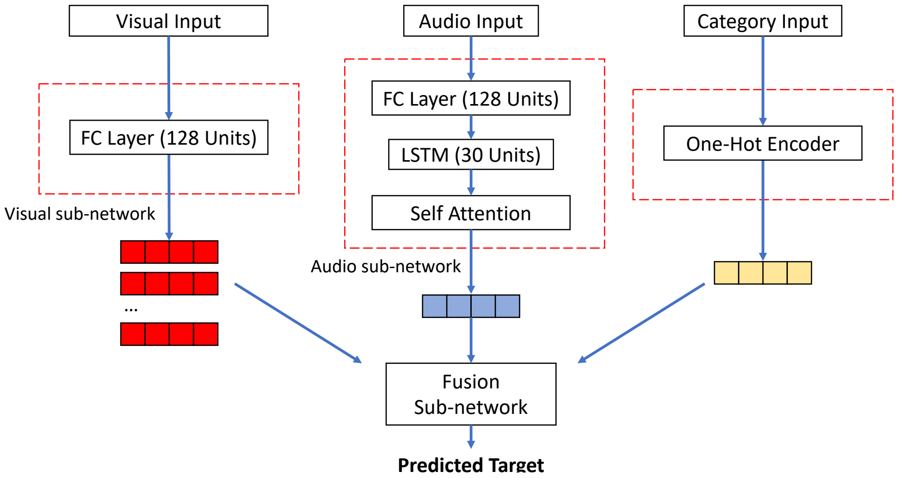


Figure 4.3: Overview of the proposed framework.

4.2.3 Proposed Approaches

The overview architecture of our system is shown in Figure 4.3. It consists of two parts: single modality sub-networks (visual and audio) and fusion sub-network. In this study, we train all these sub-networks jointly.

Feature extraction

In our study, CTR is related to the entire video, while 3-second play rate only corresponds to the content in the beginning three seconds. Therefore, we prepare the feature extraction separately for these two tasks. For CTR prediction, we utilize our in-house key-frame extractor to extract 8 visual frames from each video and keep entire audio track. For 3-second play rate prediction, we extract 3 visual frames only from the first three seconds, one frame per second and only keep audio track of the first three seconds. Extracted visual frames and audio tracks are then processed through Mobilenetv2 Sandler *et al.* (2018) and VGGish Hershey *et al.* (2017) respectively to generate visual and audio inputs. Our primary research showed that advertisement category also had impact on CTR prediction result and thus, we introduce category into our case as an extra modality. Category information (we have 19 advertisement categories) are processed by an one-hot encoder to generate category embedding.

Single Modality Sub-networks

Inputs of visual and audio modalities are processed by two sub-networks respectively. The visual sub-network has a fully-connected (FC) layer, whose parameters are shared across all frames. The output of visual sub-network is a sequence of 128-D embeddings. The audio sub-network consists of a FC layer, an uni-directional LSTM layer and a self-attention layer. The output of audio sub-network is a 128-D embeddings. Numbers of neurons in each layer are shown in Figure 4.3. For each video, we have several frames for visual input, while we have only one embedding for audio input and one embedding for category. Therefore, audio embedding and category embedding are repeated to match the frame count in each video. These collected embeddings are then sent to our proposed fusion model.

Fusion Sub-network

In our study, we have two fusion approaches: baseline and self-organizing. For the baseline approach, we adopt fusion model sub-network proposed in Ti-AVC network Su *et al.* (2020) (Figure 4.4). It has two 1-D convolutional neural networks (CNN), one max-pooling layer and one FC layer to predict targets. For our proposed approach, which we name as “self-organizing" approach, we follow the following steps to learn fusion strategy from data: (1) Modify the fusion sub-network in the baseline approach. We connect input embeddings to the second CNN layer and the max-pooling layer, in addition to the first CNN layer. The output of the first CNN layer is also connected to the max-pooling layer, as shown in dashed border of Figure 4.5. It is equivalent to connecting the output of each layer to *all* of its following layers in the dashed boundary. Therefore, we name it as “all-connected"

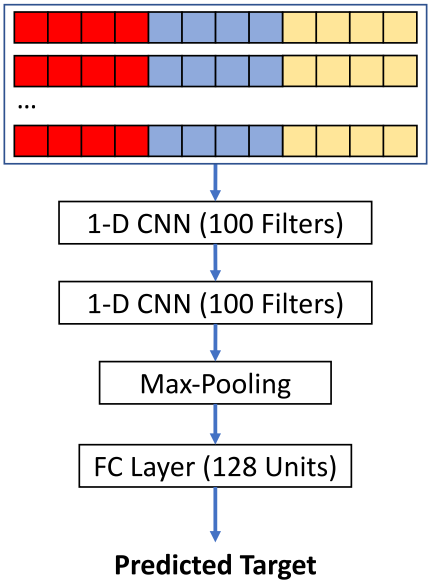


Figure 4.4: Baseline model.

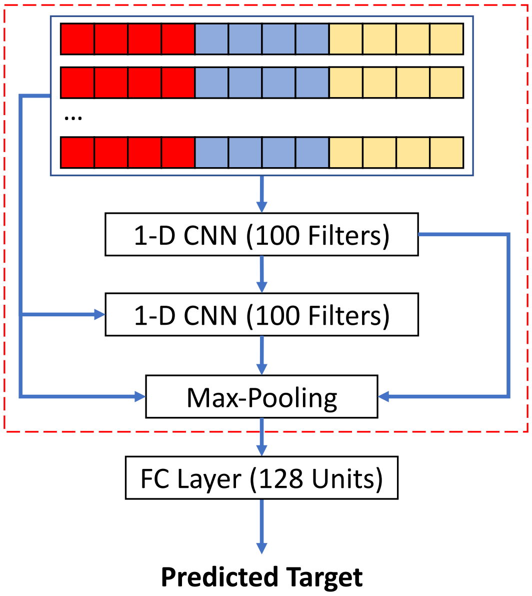


Figure 4.5: All-connected model.

fusion sub-network. (2) Optimize all-connected fusion sub-network until there is no more improvement in performance on validation set (shown in Figure 4.6). (3) Select the 5% of connections with the lowest absolute values (shown in Figure 4.6). (4) Remove connections selected in step (3) and fine-tune the sub-network (shown in Figure 4.6). (5) Repeat step (3) and step (4) until parameters number reaches a pre-defined threshold. We employ the parameter number of fusion model in baseline approach as our threshold in the experiments.

The logic behind removing connections is that connections with low absolute values indicate that they play less important role in forward propagation than others. With these connection removed, our model re-organizes information flow and learns the optimal topology. Therefore, we name our fusion model as self-organizing model. The entire procedure is data-driven and does not require manually defined rules. Our self-organizing framework is flexible and sophisticated.

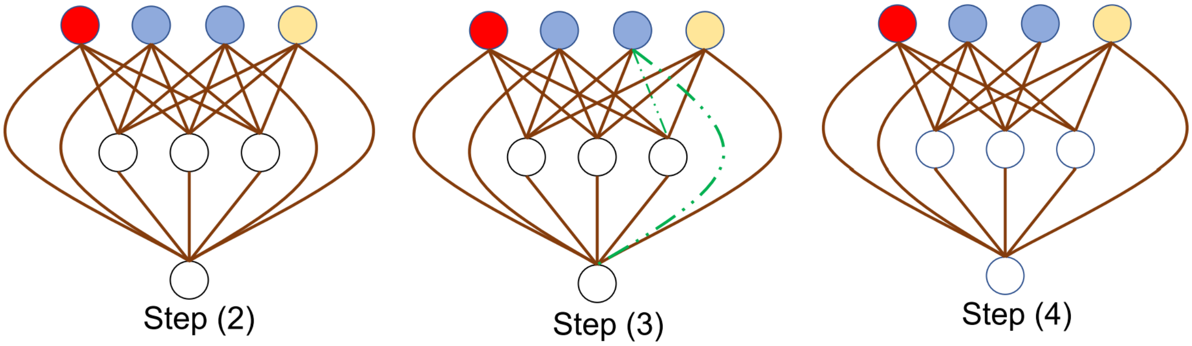


Figure 4.6: Step (2), (3) and (4) of our self-learning approach. We simplify the diagrams for illustration.

4.2.4 Experiment and Analysis

Experiment Setup

We evaluated our proposed method on CTR prediction and 3-second play rate tasks. For each task, two types of experiments were conducted: regression and classification. For the classification experiment, we uniformly binned the data into five groups (every bin had same count of sample in training set), based on the distribution of data on training set (shown in Figure 4.7 and 4.8). Here, the 1-D output layer in regression model was replaced by a 5-D softmax output layer.

Two models, which have been introduced in Section 4.2.3, were built for comparison with our proposed work. The first one was the fusion model in the baseline approach (named as “Baseline"). The other model for comparison was the fusion model in all-connected approach (named as “All-Connected"). It should be noted that “All-Connected" model has more neuron connections than “Baseline" model with the same neuron number. To make fair comparison, we trimmed its neuron number (51% of kernels in convolutional layers and neurons in dense layer) to ensure it has same parameters number as “Baseline" model. We also made the final parameters number of our proposed approach (named as

“Self-Organizing") same as the “Baseline" model. In other words, all three models had the same number of parameters.

80% and 10% of the samples were randomly selected as our training set and validation set, while the rest were used as our testing set. We adopted *Adam* as our optimizer. 0.0001 and 8 were chosen as learning rate and batch size respectively in all experiments. In all regression experiments, *mean squared error* (MSE) and *mean absolute error* (MAE) were employed as our evaluation metrics. To fairly compare all models' performance on different tasks, in addition to the two metrics above, we utilized ratio of MAE to Average ground truth (named as “MAR" in our study) as our main evaluation metric. *Mean absolute percentage error* (MAPE) was not used in our study, as it was more likely to be effected by outlier samples, which had low MAE but introduced extremely high MAPE. In classification experiments, the accuracy of all models were summarized.

Experiment Results and Analysis

The results of all regression experiments are listed in Table 4.3 and 4.4. In CTR regression experiment, our proposed Self-Organizing model beats Baseline model and All-Connected model by 0.9% and 6.0% respectively (absolute difference). In 3-second play rate regression experiment, our proposed Self-Organizing model outperforms Baseline model and All-Connected model by 0.5% and 0.3% respectively (absolute difference). We notice that MAE of Self-Organizing model is 2.5% lower than Baseline model and 1.8% lower than All-Connected model. As shown in Figure 4.7 and 4.8, the CTR distribution follows an heavy-tail distribution, while 3-second play rate follows a normal distribution. In both types of distribution, our proposed model achieves the best performance, indicating that it has strong flexibility and generalization ability.

Table 4.5 summarizes classification experiment results. In CTR classification experiment,

Model	MSE	MAE	MAR
Baseline	$1.87e^{-05}$	$2.26e^{-03}$	35.5%
All-Connected	$2.00e^{-05}$	2.58^{e-03}	40.6%
Self-Organizing	$\mathbf{1.77{\times}10^{-5}}$	$\mathbf{2.20{\times}10^{-3}}$	**34.6%**

Table 4.3: Summary of experimental results of CTR prediction.

Model	MSE	MAE	MAR
Baseline	0.0152	0.0744	17.6%
All-Connected	0.0148	0.0738	17.4%
Self-Organizing	**0.0143**	**0.0725**	**17.1%**

Table 4.4: Summary of experimental results of 3-second play rate prediction.

CTR		3-second Play Rate	
Model	Acc	Model	Acc
Baseline	61.3%	Baseline	61.4%
All-Connected	63.5%	All-Connected	61.5%
Self-Organizing	**66.3%**	**Self-Organizing**	**62.8%**

Table 4.5: Summary of experimental results of multi-class classification. “Acc" here refers to accuracy.

our proposed model outperforms Baseline model and All-Connected model by 5.0% and 2.8% respectively (absolute difference). In 3-second play rate classification experiment, our proposed model outperforms Baseline model and All-Connected model by 1.4% and 1.3% respectively (absolute difference). We note that classification is a task with coarser granularity compared with regression. It shows that our proposed Self-Organizing model outperforms other baseline models in all granularities.

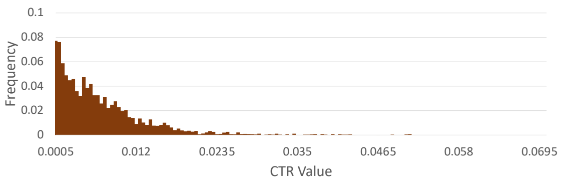


Figure 4.7: CTR distribution.

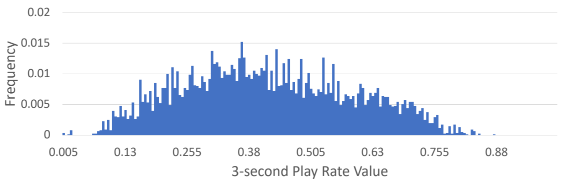


Figure 4.8: 3-second play rate distribution.

4.2.5 Discussion

In this study, we propose a self-organizing approach, which can learn the optimal topology of neural network architecture in a data-driven way. Unlike previous approaches, our proposed method does not require prior knowledge or assumption about relationship among modalities. It provides more flexibility in handling tasks related to UGVs, which contain complex and complicated information. Also, our proposed method is able to predict CTR and 3-second play rate directly from video inputs. Our experimental results reveal that our proposed method successfully predict user behaviors and outperforms all other models for comparison.

Chapter 5

Conclusion and future direction

In this paper we discussed the exploration and application of data with complex structures. In bio-informatics field, we proposed a convolutional self-attention based model to capture the hidden information within DNA sequences and motifs. This model also shows its potential in the for biological interpretation. In the sales forecasting field, we proposed a word-to-vector based data processing pipeline to convert the complicated online clickstream data into vectors, then we ran experiments on different models. Besides, we further tested the potential improvement of the multitask learning with graph attention network. In multimedia field, our contribution includes the learning of single modality and the learning of across multiple modalities. For single modality, we proposed an emsemble end-to-end spoken language model to learn the information from spoken language audio. Then for multimodality, we applied a themes informed audio video correspondence learning model, along with a self-adapted learning algorithm to optimize the information flow within the model. In the future, we have three possible directions. The first direction is the quantitative analysis of different architectures. A complicated model is more likely to get overfitting during training, which makes it hard to converge on noisy data. A quantitative analysis for the selection of regularization model architecture will be very significant during application. Another direction is multitask learning. There are lots of datasets with similar structure. When the sample size is limited, transfer knowledge from a large dataset to a smaller dataset will be very helpful. Besides, how to separate the shared information and the task specific information can further improve the model performance. The last direction is to capture the information from multiple domains. Different from multitask learning, a very popular topic now is combining the inputs from multiple domains and make them contribute to the same target. Our researches have demonstrated that new modality can help us achieve a better performance on multimedia services. Based on that, we may extend the contribution to more modalities and more fields.

BIBLIOGRAPHY

BIBLIOGRAPHY

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

Afouras, T., Chung, J. S., Senior, A., Vinyals, O., and Zisserman, A. (2018). Deep audio-visual speech recognition. *IEEE transactions on pattern analysis and machine intelligence*.

Alipanahi, B., Delong, A., Weirauch, M. T., and Frey, B. J. (2015). Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nat Biotechnol*, **33**(8), 831–838.

Arandjelovic, R. and Zisserman, A. (2017). Look, listen and learn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 609–617.

Arandjelovic, R. and Zisserman, A. (2018). Objects that sound. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 435–451.

Aytar, Y., Vondrick, C., and Torralba, A. (2016). Soundnet: Learning sound representations from unlabeled video. In *Advances in neural information processing systems*, pages 892–900.

Baltrušaitis, T., Ahuja, C., and Morency, L.-P. (2018). Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence*, **41**(2), 423–443.

Benveniste, D., Sonntag, H.-J., Sanguinetti, G., and Sproul, D. (2014). Transcription factor binding predicts histone modifications in human cell lines. *Proceedings of the National Academy of Sciences*, **111**(37), 13367–13372.

Berkovec, J. (1985). New car sales and used car stocks: A model of the automobile market. *The Rand Journal of Economics*, pages 195–214.

Brühl, B., Hülsmann, M., Borscheid, D., Friedrich, C. M., and Reith, D. (2009). A sales forecast model for the german automobile market based on time series analysis and data mining methods. In *Industrial Conference on Data Mining*, pages 146–160. Springer.

Chase, C. W. (2013). *Demand-driven forecasting: a structured approach to forecasting*. John Wiley & Sons.

Chen, H., Ding, G., Liu, X., Lin, Z., Liu, J., and Han, J. (2020). Imram: Iterative matching with recurrent attention memory for cross-modal image-text retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12655–12663.

Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., *et al.* (2016). Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*, pages 7–10.

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

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

Chung, S.-W., Chung, J. S., and Kang, H.-G. (2019). Perfect match: Improved cross-modal embeddings for audio-visual synchronisation. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3965–3969. IEEE.

Cox, A. (2018). *2018 Car Buyer Journey Study. Cox Automotive*. Cox, Automotive.

Cramer, J., Wu, H.-H., Salamon, J., and Bello, J. P. (2019). Look, listen, and learn more: Design choices for deep audio embeddings. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3852–3856. IEEE.

Diba, A., Fayyaz, M., Sharma, V., Paluri, M., Gall, J., Stiefelhagen, R., and Van Gool, L. (2020). Large scale holistic video understanding. In *European Conference on Computer Vision*, pages 593–610. Springer.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9, pages 249–256. PMLR.

Hershey, S., Chaudhuri, S., Ellis, D. P., Gemmeke, J. F., Jansen, A., Moore, R. C., Plakal, M., Platt, D., Saurous, R. A., Seybold, B., *et al.* (2017). Cnn architectures for large-scale audio classification. In *2017 ieee international conference on acoustics, speech and signal processing (ICASSP)*, pages 131–135. IEEE.

Hochreiter, S. and Schmidhuber, J. (1997a). Long short-term memory. *Neural computation*, **9**(8), 1735–1780.

Hochreiter, S. and Schmidhuber, J. (1997b). Long short-term memory. *Neural computation*, **9**(8), 1735–1780.

Hu, D., Nie, F., and Li, X. (2019). Deep multimodal clustering for unsupervised audiovisual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9248–9257.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.

Huang, Q., Xiong, Y., Rao, A., Wang, J., and Lin, D. (2020). Movienet: A holistic dataset for movie understanding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 709–727. Springer.

Huang, T. and Van Mieghem, J. A. (2014). Clickstream data and inventory management: Model and empirical analysis. *Production and Operations Management*, **23**(3), 333–347.

Kayapinar Kaya, S. and Yildirim, Ö. (2020). A prediction model for automobile sales in turkey using deep neural networks. *Journal of Industrial Engineering (Turkish Chamber of Mechanical Engineers)*, **31**(1).

Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Kulmanov, M., Khan, M. A., Hoehndorf, R., and Wren, J. (2018). DeepGO: predicting protein functions from sequence and interactions using a deep ontology-aware classifier. *Bioinformatics*, **34**(4), 660–668.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, **521**(7553), 436–444.

Lewandowski, R. (1974). Prognose-und informationssysteme und ihre anwendungen bd. 1. *Berlin ua*.

Li, B. and Kumar, A. (2019). Query by video: Cross-modal music retrieval. In *ISMIR*, pages 604–611.

Liu, C., Tang, T., Lv, K., and Wang, M. (2018). Multi-feature based emotion recognition for video clips. In *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, pages 630–634.

Lu, R., Duan, Z., and Zhang, C. (2019). Audio–visual deep clustering for speech separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, **27**(11), 1697–1712.

Lu, W., Chen, S., Li, K., and Lakshmanan, L. V. (2014). Show me the money: dynamic recommendations for revenue maximization. *Proceedings of the VLDB Endowment*, **7**(14), 1785–1796.

Mikolov, T., Karafiát, M., Burget, L., Černockỳ, J., and Khudanpur, S. (2010). Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv:1310.4546*.

Monfort, M., Andonian, A., Zhou, B., Ramakrishnan, K., Bargal, S. A., Yan, T., Brown, L., Fan, Q., Gutfreund, D., Vondrick, C., *et al.* (2019). Moments in time dataset: one million videos for event understanding. *IEEE transactions on pattern analysis and machine intelligence*, **42**(2), 502–508.

Noroozi, F., Marjanovic, M., Njegus, A., Escalera, S., and Anbarjafari, G. (2017). Audio-visual emotion recognition in video clips. *IEEE Transactions on Affective Computing*, **10**(1), 60–75.

Quang, D. and Xie, X. (2016). Danq: a hybrid convolutional and recurrent deep neural network for quantifying the function of dna sequences. *Nucleic acids research*, **44**(11), e107–e107.

Quang, D. and Xie, X. (2019). Factornet: a deep learning framework for predicting cell type specific transcription factor binding from nucleotide-resolution sequential data. *Methods*, **166**, 40–47.

Ritchie, M. E., Phipson, B., Wu, D., Hu, Y., Law, C. W., Shi, W., and Smyth, G. K. (2015). limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Res*, **43**(7), e47.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520.

Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, **45**(11), 2673–2681.

Slattery, M., Zhou, T., Yang, L., Machado, A. C. D., Gordân, R., and Rohs, R. (2014). Absence of a simple code: how transcription factors read the genome. *Trends in biochemical sciences*, **39**(9), 381–399.

Song, Q., Cheng, D., Zhou, H., Yang, J., Tian, Y., and Hu, X. (2020). Towards automated neural interaction discovery for click-through rate prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 945–955.

Su, R., Tao, F., Liu, X., Mei, H. W. X., Duan, Z., Yuan, L., Liu, J., and Xie, Y. (2020). Themes inferred audio-visual correspondence learning. *arXiv preprint arXiv:2009.06573*.

Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., and Liu, C. (2018). A survey on deep transfer learning. *CoRR*, **abs/1808.01974**.

Tao, F. and Busso, C. (2018a). Aligning audiovisual features for audiovisual speech recognition. In *IEEE International Conference on Multimedia and Expo (ICME 2018)*, pages 1–6, San Diego, CA, USA.

Tao, F. and Busso, C. (2018b). End-to-end audiovisual speech activity detection with bimodal recurrent neural models. *ArXiv e-prints (arXiv:1809.04553)*, pages 1–11.

Tao, F. and Busso, C. (2018c). Gating neural network for large vocabulary audiovisual speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, **26**(7), 1286–1298.

Tao, F. and Busso, C. (2019). End-to-end audiovisual speech activity detection with bimodal recurrent neural models. *Speech Communication*, **113**, 25–35.

Tao, F. and Busso, C. (2020). End-to-end audiovisual speech recognition system with multitask learning. *IEEE Transactions on Multimedia*.

Tao, F., Liu, G., and Zhao, Q. (2018). An ensemble framework of voice-based emotion recognition system for films and tv programs. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6209–6213. IEEE.

Uygun, S., Seddon, A. E., Azodi, C. B., and Shiu, S.-H. (2017). Predictive models of spatial transcriptional response to high salinity. *Plant Physiology*, **174**(1), 450–464.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. (2017). Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Verma, G., Dhekane, E. G., and Guha, T. (2019). Learning affective correspondence between music and image. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3975–3979. IEEE.

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and computing*, **17**(4), 395–416.

Wang, J., Xu, Q., Wang, Q., Lyu, Z., Chen, J., and Xu, W. (2019a). Mmctr: A multi-task model for short video ctr prediction with multi-modal video content features. In *2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 679–682. IEEE.

Wang, M., Tai, C., E, W., and Wei, L. (2018). Define: deep convolutional neural networks accurately quantify intensities of transcription factor-dna binding and facilitate evaluation of functional non-coding variants. *Nucleic acids research*, **46**(11), e69–e69.

Wang, R., Huang, H., Zhang, X., Ma, J., and Zheng, A. (2019b). A novel distance learning for elastic cross-modal audio-visual matching. In *2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 300–305. IEEE.

Wang, X., Du, Y., Zhang, L., Li, X., Zhang, M., and Dong, J. (2019c). Exploring content-based video relevance for video click-through rate prediction. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2602–2606.

Wang, Z., Ma, Y., Liu, Z., and Tang, J. (2019d). R-transformer: Recurrent neural network enhanced transformer. *arXiv preprint arXiv:1907.05572*.

Weingarten-Gabbay, S. and Segal, E. (2014). The grammar of transcriptional regulation. *Hum Genet*, **133**(6), 701–711.

Weiss, K., Khoshgoftaar, T. M., and Wang, D. (2016). A survey of transfer learning. *Journal of Big data*, **3**(1), 1–40.

Whitaker, J. W., Chen, Z., and Wang, W. (2015). Predicting the human epigenome from dna motifs. *Nature methods*, **12**(3), 265.

Wilkins, O., Hafemeister, C., Plessis, A., Holloway-Phillips, M.-M., Pham, G. M., Nicotra, A. B., Gregorio, G. B., Jagadish, S. K., Septiningsih, E. M., Bonneau, R., and Purugganan, M. (2016). Egrins (environmental gene regulatory influence networks) in rice that function in the response to water deficit, high temperature, and agricultural environments. *The Plant Cell*, **28**(10), 2365–2384.

Wu, J., Xu, Y., Zhang, S.-X., Chen, L.-W., Yu, M., Xie, L., and Yu, D. (2019a). Time domain audio visual speech separation. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 667–673. IEEE.

Wu, Y., Zhu, L., Yan, Y., and Yang, Y. (2019b). Dual attention matching for audio-visual event localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6292–6300.

Xu, C., Peak, D., and Prybutok, V. (2015). A customer value, satisfaction, and loyalty perspective of mobile application recommendations. *Decision Support Systems*, **79**, 171–183.

Yu, J., Zhang, S.-X., Wu, J., Ghorbani, S., Wu, B., Kang, S., Liu, S., Liu, X., Meng, H., and Yu, D. (2020). Audio-visual recognition of overlapped speech for the lrs2 dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6984–6988. IEEE.

Zhao, H., Gan, C., Rouditchenko, A., Vondrick, C., McDermott, J., and Torralba, A. (2018). The sound of pixels. In *Proceedings of the European conference on computer vision (ECCV)*, pages 570–586.

Zhou, J. and Troyanskaya, O. G. (2015). Predicting effects of noncoding variants with deep learning–based sequence model. *Nature methods*, **12**(10), 931–934.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., and Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI Open*, **1**, 57–81.

Zhou, M., Ding, Z., Tang, J., and Yin, D. (2018). Micro behaviors: A new perspective in e-commerce recommender systems. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pages 727–735.

Zhu, H., Luo, M.-D., Wang, R., Zheng, A.-H., and He, R. (2021a). Deep audio-visual learning: A survey. *International Journal of Automation and Computing*, pages 1–26.

Zhu, Y., Wu, Y., Latapie, H., Yang, Y., and Yan, Y. (2021b). Learning audio-visual correlations from variational cross-modal generation. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4300–4304. IEEE.