CS245 Project Report
COVID-19 Pandemic Prediction with DGNN

Team Ideas: Chenyang Wang, Danfeng Guo, Haochen Yin
Huiling Huang, Panqiu Tang, Yijing Zhou

December 15, 2020

Abstract

In 2020, COVID-19 (COV19) has been widely spread and become a pandemic. Controlling the population mobility and identifying groups of people who are likely to be infected is important to stop it from spreading. We implemented a model based on Graphical Neural Networks (GNN) and Long-Short Term Memory (LSTM) to predict regional daily new infected cases. Our best model achieves 134 mean absolute error for each state in the US.

Contents

1 Introduction **3**

2 Related Works **4**

3 Approach and Implementation **7**
3.1 Problem Statement 7
3.2 Prediction Models 7

4 Experiments and Results **12**
4.1 Dataset 12
4.1.1 Preprocessing 12
4.1.2 Data Analysis 14
4.1.3 Retrospection 15
4.2 Experiments 16
4.3 Results 16

5 Discussion and Future Work **19**
5.1 Data 19
5.2 Model 19
5.3 Evaluation 20

6 Work Distribution **21**

1 Introduction

COV19 is an infectious disease caused by SARS-CoV-2. The disease was first discovered in Wuhan City, China at the end of 2019. Since then, the disease has spread rapidly in many countries around the world. The recent outbreak of COVID-19 has affected the entire world: millions of people died, tens of millions have been infected, and hundreds of millions of people's lives have been drastically changed. COV19 has already become the biggest challenge to global healthcare system. To prevent the spread of COV19, it is important to identify those who are likely to be infected and conduct test on them. A common approach is to identify high-risk groups from the moving paths of individuals. However, the availability and complexity of mobility data brings challenge to the task.

Recently, models powered by data science and neural networks have been widely explored to predict the pandemic growth. Those researches provide inspiration on building COV19 prediction models. Recent approaches in machine learning have introduced various models to predict new cases with the intention of combating the spread of the disease, assisting in the policy making decision process, and determining the allocation of health resources. Graph-based approach is an intuitive way to model the problem of pandemic forecasting based on regional and connectivity information. Given that COV19 is transmissive, mobility within and between regions plays an important role in the spread of the virus. This information can be modeled with a spatial graph, with nodes representing regions and edges representing hyperparameters such as mobility. Furthermore, the progression of the pandemic can be modeled by temporal graphs with edges between nodes that represent the same location but at different time stamps. Building on this, we can construct graph neural networks with dynamic features (such as number of cases) to learn the pattern of the spread of the virus. Recent GNN approaches have achieved success in spatio-temporal influenza forecasting [1]. Graph neural networks have been especially useful in this case since they are able to utilize adjacent node information to inform the future hidden states for a node.

In this project, we implement a series of models that utilize both spatial and temporal information in dynamic GNN to capture the pattern of the spread of the disease. We train the models using two datasets, SafeGraph Mobility Data and CDC Pandemic Trackers. The SafeGraph Mobility Dataset contains a rich set of features that allow us to model mobility and proximity on both temporal and spatial scales. The CDC dataset contains the number of daily confirmed cases, probable cases, and deaths in different regions across the U.S.. We test our models on various time periods and compare the predicted number of cases against the true number of cases. The model that combines a Message Passing Neural Network (MPNN) and Long-Short Term Memory network (LSTM) with a self-attention mechanism achieves the best performance with a steady and small range of errors as time increases.

The rest of the report is organized as follows. Section 2 presents related works and models also about the challenge of pandemic forecasting for the cases of COVID-19. Section 3 discusses our problem in details, methods and implementation with respect to formulating dynamic graph neural networks and different models. Section 4 describes our datasets, experiments, and results in COVID-19 forecasting. Section 5 summarizes our work, discuss potential improvements and future work, and also states individual contributions.

2 Related Works

When considering any infectious diseases, the first and simple model to be considered is the SIR model, which is a compartmental model in epidemiology. It is a model proposed by Kermack and McKendrick back to the year of 1927. The model consists of three compartments: S for the number of susceptible, I for the number of infectious, and R for the number of recovered or deceased (or immune) individuals, where the parameters β, σ and γ are used to describe the different transition rates. The dynamics of an epidemic, for example, the flu, are often much faster than the dynamics of birth and death, therefore, birth and death are often omitted in simple compartmental models. However, how to select these transfer parameters is a critical problem in establishing the model. These parameters are not selected and set by us, but are automatically generated by the deep learning models, which is currently an ongoing research area given the situation of the COVID-19 pandemic.

Given the nature of the COVID-19, which is a kind of infectious disease, and the availability of mobility data that current technology provided to us, it is suitable to combine all information and formulate it into a graph problem, where each node can represent a certain area with some features stored and the weighted edges can be defined by the mobility data. And indeed, many studies on GNNs have already been conducted Among them, we are interested in the application of GNNs on solving problems like Social Influence Prediction and Disease Spread Prediction.

For example, Trivedi et al. proposed a learning framework called DyRep [2], which is specifically designed for dynamic graphs. DyRep captures the underlying dynamics of node interactions, predicts dynamic links and performs some time-related tasks. DyRep investigates two major questions for representation learning over dynamic graphs. The first one is what can serve as an elegant model for dynamic processes over graphs. The second one is how can we leverage such model to learn dynamic node representations that are effectively able to capture evolving graph information over time. Dynamic graphs can be modeled in two different dynamic processes. Growing or shrinking of the nodes and edges over time is considered as topological evolution. Activities between nodes that may or may not be connected is considered as node interactions. With the model defined, the next challenge is to effectively model and learn representations that capture the key dynamical properties of such system with highly nonlinear evolution. DyRep has a general framework trying to tackle these two major challenges. The basic framework of DyRep can be summarized into three steps. In the first step, a double scale deep temporal point process is built in order to obtain the temporal dynamics if the two observed processes are in continuous-time domain. Then a conditional intensity function of the temporal point process is parameterized with a deep inductive representation network that learns some functions to compute the node representations. The last step is to propose a new Temporal Attention Mechanism to combine the structural and temporal components together so that the dynamics can be captured over time. This framework is trained by an end-to-end unsupervised training procedure.

Qiu et al. proposed a Deep Learning framework called DeepInf [3] that utilize GNN techniques to learn node's latent feature representation for predicting social influence. The definition of social influence can be very broad, and the paper defined it as the phenomenon that a person's emotions, opinions, or behaviors are affecting other people. And this concept is critical in the applications such as online recommendation and advertising. DeepInf is a framework that can characterize, understand, and quantify the underlying mechanisms and dynamics of social influence.

In particular, it is to predict the action of a user given the action of the user's near neighbors and local structural information. There are three major components of this framework: network embedding, graph convolution, and graph attention mechanism. The basic idea of this framework is to first sample a user (node)'s local neighbors by random walks with restart, and then after obtaining the local network, graph convolution and attention techniques are used to learn latent predictive signals. Obviously, this is a graph problem, and each user can be view as a node in the graph, and edges connect users that can have influence on each other. The first step is to do the sampling. This is done by using Random Walks with Restart (RWR) with the assumption that active nodes will tend to influence their neighbors more likely than inactive nodes. The RWR starts with either one of the ego users or its active neighbors, and then iteratively traverse to its neighbors with a probability proportional to the weight of the edges. The RWR stops when a pre-defined number of vertices have been visited. Then the raw input is fed into a neural network. First, the input is fed into the embedding layer, which uses a pre-trained model to convert a node into a lower dimensional representation vector. And then normalization is used to avoid overfitting during training. After that, the input layer constructs a feature vector for each node. Then it is fed into a GNN or a Graph Attention Network (GAT). And then the output layer outputs a 2D representation for each user, and it is used to compare to the ground truth to optimize the loss function. These models and frameworks provide us some general ideas and techniques for node's representation learning as well as learning the dynamics of the graph.

In terms of Disease Spread Prediction, Deng et al. proposed a GNN framework for epidemic forecasting called ColaGNN [1], that first learns latent representation for each node representing each location using Recurrent Neural Networks (RNN), then use these learnt representations to derive an attention matrix that captures the influence and dynamics between locations. However, this model has only been tested on Influenza-Like Illness (ILI) in US and Japan and has not yet been evaluated for this COV19 pandemic. Another model called STAN [4] was proposed. It designs nodes and edges based on demographical and geographical similarities between regions. It also takes in various kinds of data including recovered and death cases, local medical resource and disease data, and integrated pandemic transmission patterns with deep learning models. Each node represents a region and has static and dynamic attributes from real world data, while edges represent demographical and geographical similarities. The model incorporates attention mechanism into GNN for interaction between neighbors. Predictions are made in a fixed time range, while keeping physical constraints that underly transmission dynamics in the real world. The model consists of a GNN for spatial information, an RNN that captures temporal information, and physical law constraints for different time ranges. A two-layer GNN captures spatial features from the latest data and part of historical data within a sliding window. Graph attention mechanism (GAT) can learn hidden embeddings of nodes and calculates attention coefficients, followed by self-attention and aggregation for each node. This can model the knowledge that different regions can have different infectious impact. Then the result is aggregated from all nodes and fed into Gate Recurrent Unit (GRU) to produce an embedding that contain both spatial and temporal information. The physical constraints include predicting trans- mission/recovery rates over time with MLP, number of infected/recovered cases, and physics constraints loss from calculated data and SIR differential equations. STAN achieved better performance than epidemiological modeling methods and deep learning methods however with some limitations, one limitation of the

network is the sliding window prediction which requires more accurate data input, which can be further solved by dynamic smoothing to the data, and another limitation is that the physical constraint models may be too simple for the realistic situations, and it can be improved by incorporating more population groups and transmission equations, similar to improving the SIR model.

We use MPNN-LSTM model. The MPNN structure has graphical convolutional layers that have the advantage of encoding the information among different regions. The LSTM structure [5] has already shown excellent performance in Natural Language Processign (NLP) tasks for dealing with sequential data. We also add a self-attention mechanism to LSTM to make the model focus on the data of the most related dates.

3 Approach and Implementation

3.1 Problem Statement

Graph Formulation Regions at the same granularity level (eg: states, counties) are represented as nodes in graphs. Each node has its daily confirmed cases as its feature. The edges between nodes indicate the visitor flow and the edge feature is the number of visitors. This is represented as an adjacency map whose values are the mobility between nodes. Then to represent temporal information, different layers of graphs are joined by edges between nodes that represent the same location but different timestamps. For example, there will be a temporal edge between the node (California at March 8) and the node (California at March 9). The dynamic attributes in the graph is captured in a feature matrix along with static attributes, and the attributes in each step are updated.

In our project, we perform COV19 prediction on state level. The graph nodes are the states of the U.S.. There are 51 nodes in total. (including Washington D.C.) For each pair of nodes, there are one forward edge and one backward edge indicating the visitors going to and back from the target state. The node features are the daily newly confirmed COV19 cases.

Task Formulation We denote X as the daily new cases of consecutive W days, and $X \in \mathbf{R}^{W*\times N\times 1}$, where we set W as 7, N is the number of states in the US including Washington DC and equal to 51. A is the corresponding daily mobility data between states and $A \in \mathbf{R}^{W\times 51\times 51}$. For example, A_{wij} indicates the number of visitors from state i to state j on day w. The task is to predict the daily new cases of each state on the following 6th, 10th, 14th, 18th and 22th day. The reason for our choice of dates is that, the average time from being infected to having symptoms is 14 days. Hence, we suppose our input data is more closely related the results around 14 days later. Models that try to predict cases on days too early or too late may not return a satisfactory performance. We expect models trained to predict the cases of the 6th and 22th day show a decrease on performance.

3.2 Prediction Models

MPNN [6] Given a series of Graphs at different timestamps, we will update the node representations in each layer based on its adjacency information and recent past temporal information. MPNN contains neighborhood aggregation layers that can update each layer of graphs based on the normalized weight adjacency matrices following Kipf and Welling [7]. When computing node features, we want to incorporate temporal information by summing nodes in the spatial range in a sliding window of specified time range. This model is applied to all layers in a graph that differ by time, with unique adjacency matrices and node representation matrices for layers, and shared weights between layers. The model captures more global information as the number of neighborhood aggregation layers increases. However, there should be a balance between maintaining local and global information. The MPNN we use is the same as [6], which is shown in Fig. 1. It makes use of graphical convolutional layers (MPNN layer). The node representation matrices are concatenated to create skip connections between each layer and the output layer. This will allow the network to encode multi-scale structural information. Then the output of the network is followed by a ReLU.

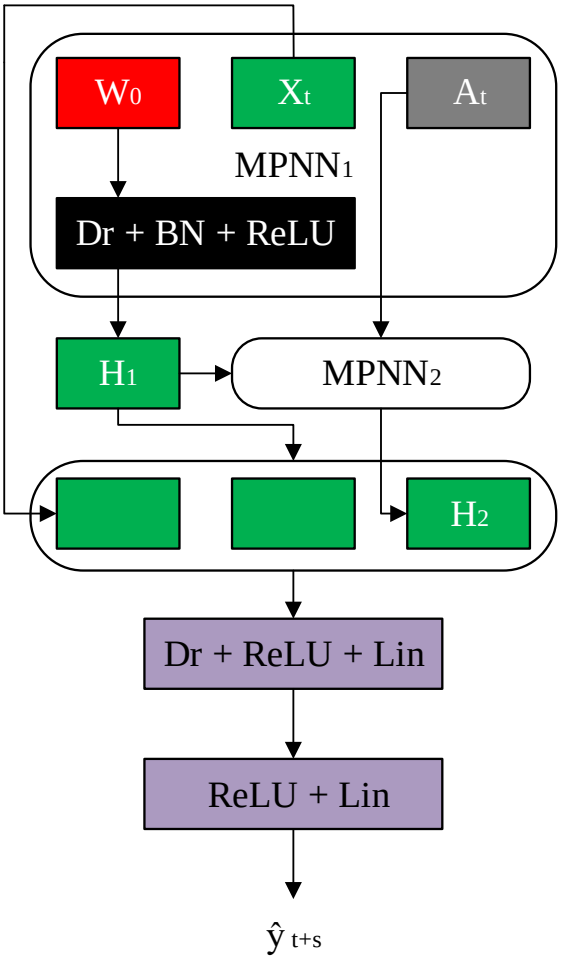


Figure 1: MPNN Network Structure

LSTM LSTM is suitable for sequential data and has been widely applied to NLP tasks. Compared with traditional RNNs, LSTM addresses the vanishing gradient problem by introducing a forget gate to control the information passing from long distance.

MPNN+LSTM The input data can be seen as a sequence of graphs. A MPNN module can be applied first to encode information among nodes within each graph. Then, LSTM module can be used to process the information across the whole sequence. The details MPNN-LSTM model are in Fig. 3. Daily new cases and mobility data of each day are separately processed by two MPNN layers. The MPNN outputs then form a sequence and are fed into two LSTM layers. We added a skip connection path after LSTM. The hidden features of two LSTM layers are concatenated with original daily new cases. The concatenated features are used to perform the regression task.

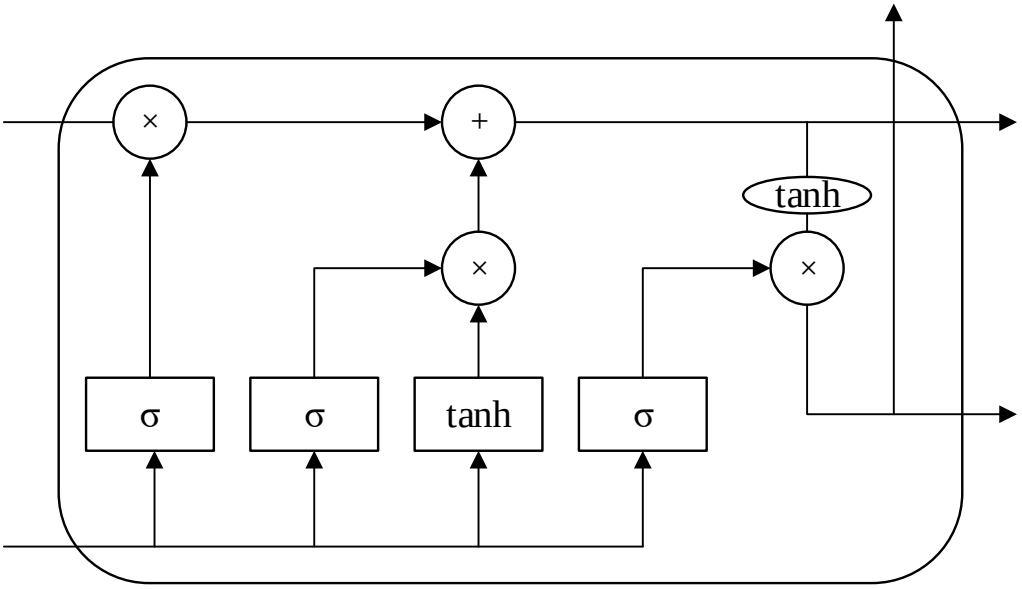


Figure 2: Details of LSTM structure

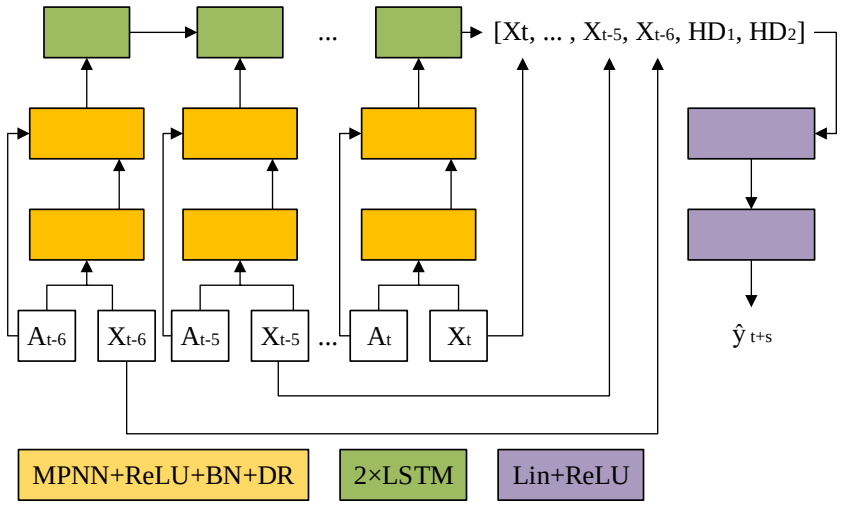


Figure 3: MPNN+LSTM

MPNN+LSTM with attention Since the duration from one gets infected to one gets positive test varies, it is not necessary that the information on the nearest day has the most importance. A self-attention mechanism [8] could be applied to traditional LSTM structure to make the model focus on the important tokens within the sequence. As shown in Fig. 4 The hidden features used to perform regression are represented as a weighted sum of hidden features of all sequence.

Transformer [9] Since its development in 2017, models based on transformer has been widely applied to numerous NLP tasks and become the state-of-art. Compared with traditional recurrent structures, transformer can be parallelized and consumes less resources. We applied transformer layers after MPNN and built a MPNN+TSFM network. The structure is shown in Fig. 5. The MPNN output is fed into transformer layers. Given transformer loses the order of sequence, positional

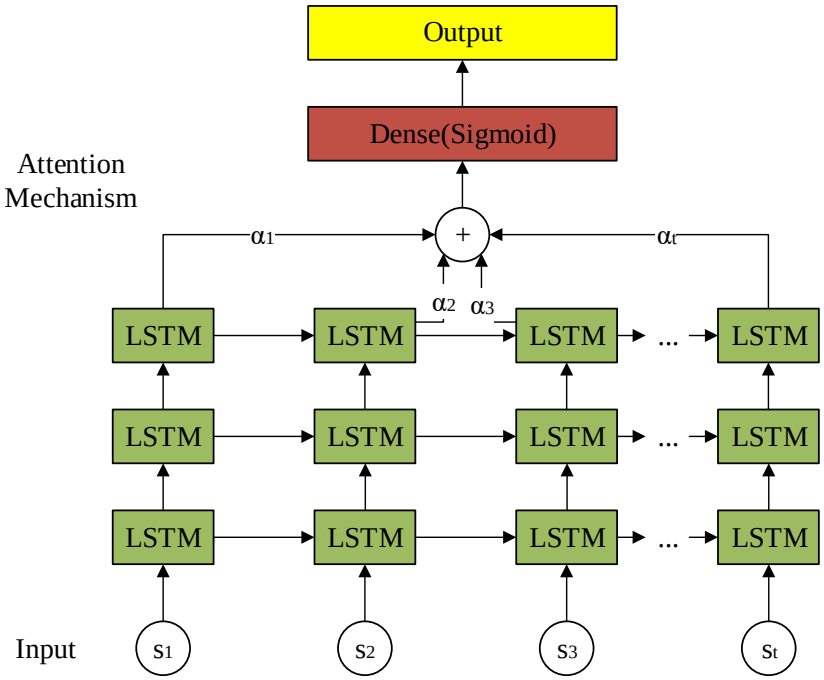


Figure 4: Structure of LSTM with attention.

encoding needs to be added to the input. Transformer consists of two parts, multi-head attention module and residual feed-forward module. The details of multi-head attention is included in Fig. 6. We introduced the concept of query (Q), key (K) and value (V) mentioned in [9]. Incoming MPNN features with positional encoding pass through different linear layers to become Q, K and V. Q and K are used to compute the softmax weights and the output is a weighted sum of values. This module can be repeated multiple times such that we have multi-head attention structure.

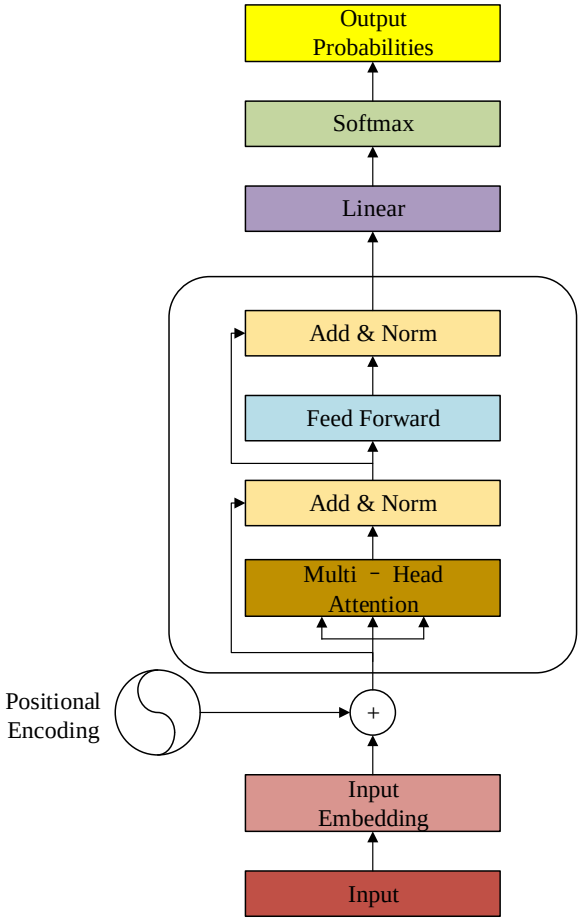


Figure 5: Structure of Transformer

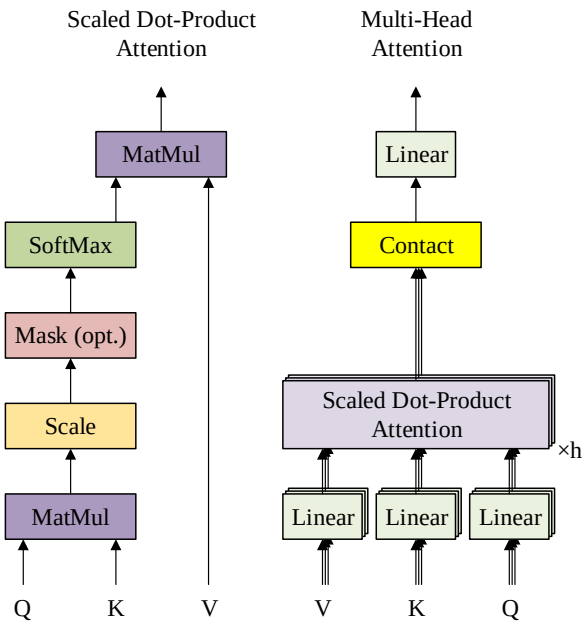


Figure 6: Multi-head attention

4 Experiments and Results

4.1 Dataset

For our model, there are two sources of datasets. We derive our measures for mobility from SafeGraph Mobility dataset, while the main COVID-19 related statistics is directly obtained from CDC Pandemic data tracker.

The SafeGraph mobility dataset is a set of data containing rich mobility information at different granularity. Our models use two major sets of data to estimate the *internal mobility* and *external mobility* of each U.S. states.

Internal mobility refers to the amount of traffic circulation inside the state. For instance, if residents in a certain area decided to leave their residences and went to public areas, we would like to consider such action as a contributing factor to *internal mobility*. *External mobility* means the level of population movement from one state to another. For instance, during holidays such as Thanksgiving Day, we would expect a large increase in external mobility for each state, as people would travel from their currently residing states to their home states to reunite with their families.

The CDC Pandemic data tracker provides key statistics that can be used as attributes for each state. In this dataset, it reports new COVID-19 related cases and deaths at daily basis. If the state consents to release certain information, it would also include statistics such as probable COVID-19 related cases and deaths for that state. Probable COVID-19 cases/deaths are defined as cases or deaths that have COVID-19 related symptoms, but there is no laboratory evidence to confirm such cases or deaths yet.

We believe that mobility and the number of COVID-19 cases are highly related to the spread of epidemic. Under this assumption, we have processed the raw datasets accordingly. Next we will detail what we have done to pre-process the data, and how the pre-processed data supports our assumption, and what we could have done better in retrospection.

4.1.1 Preprocessing

Data transformations should be conducted to extract the mobility measures that we want from SafeGraph mobility dataset. The `main` dataset provides us abundant information regarding points of interest (POIs) in each censor block group. An *POI* can be a McDonald's in Phenix City, AL. We utilized three POI attributes, **region**, **visits_by_day**, and **visitor_home_cbgs**. These attributes are particularly useful as it provides us insights about the amount of foot traffic around each POI. An example is shown in Table. 1. The feature **visits_by_day** is the daily number of visitors of the target week, and **visitor_home_cbgs** is the number of visitors grouped by each POI. We have hence made a bold hypothesis that population movement in one state can be estimated by aggregating the number of visitors of all the POIs in that state.

region	visitor_by_day	visitor_home_cbgs
TX	[19,16,11,13,23,15,13]	{"484391131041":12,"484391131042":4,"484391137...
AL	[38,26,36,42,42,22,25]	{"011130309021":13,"011130307003":12,"01113030...

Table 1: A concise example of POI data

To compute data at state level, we built a lookup table using the `home_panel`

dataset. As shown in Table. 2, this dataset provides a mapping from POI ID to its belonging state. We used this lookup table to aggregate the number of visitors in **visitor_home_cbgs** to compute the number of visitors at state level. Then the data in `main` is grouped and aggregated by states and the weekly visitor flows within and among states are computed.

Note that the data is counted by weeks. To convert it to daily base, we also aggregate the **visits_by_day** feature to compute the daily number of visitors of each state. The daily number of visitors is normalized and used as a weight vector to convert **visitor_home_cbgs** into daily basis.

date_range_start	date_range_end	state	censor block group	number of devices
2018-12-31	2019-01-07	ak	20200028232	172
2018-12-31	2019-01-07	ak	20900019002	69

Table 2: A concise example of home_panel dataset

Lastly, we create an adjacency matrix W where each non-diagonal cell estimates the *external mobility* from one state to another, while for all the diagonal entries, the value represents the *internal mobility* for that state. The overall process of data processing is shown in Fig. 7

$$W_{i,j} = \begin{cases} \textit{internal mobility of state i} \ , \ \text{i} = \text{j} \\ \textit{external mobility between state i and state j} \ , \ \textit{otherwise} \end{cases}$$

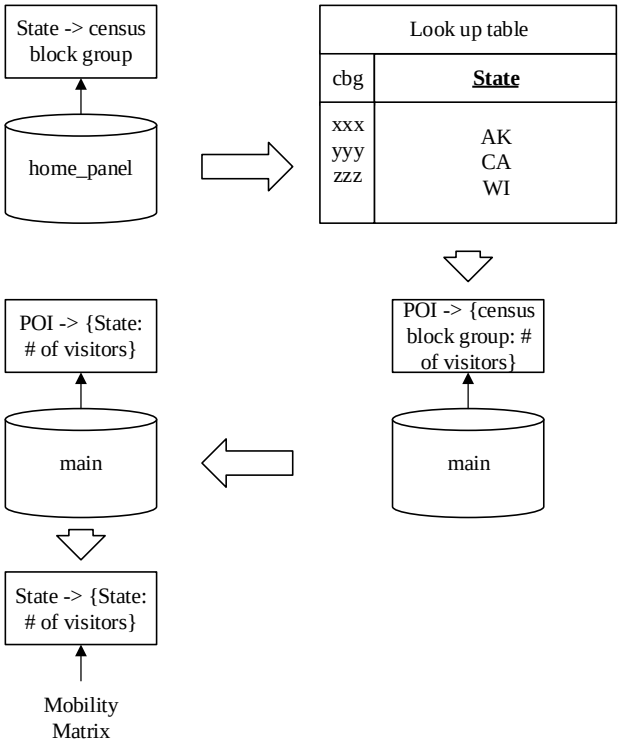


Figure 7: Illustration of the data processing steps

Data cleaning is needed for the CDC pandemic tracker dataset. In particular, we have noticed that not all states have consented to release probable and confirmed

cases (deaths) data. We just simply fill empty values with zero values. We are aware that this can make the data negatively skewed. Furthermore, to our surprises, we have also found negative values from the raw dataset. According to the official documentation, the negative values are the results of "adjustments", as daily reported values are often provisional and thus subject to change. However, it does not make sense to have negative attributes in our model, as our model uses these attributes as weights that represents in the spread of epidemic, and one cannot have negative "spread". Thus, we have replaced all the negative values with zero to adapt to our model. Finally, since the SafeGraph dataset does not have any data later than June, we have cut the CDC pandemic tracker dataset short accordingly, and normalized the data using statistics between January and June 2020.

4.1.2 Data Analysis

We have conducted some basic analysis to our derived mobility measures and checked against our hypothesis that the *mobility* indeed contributes to the spread of COVID-19. Figure. 8 is a scatter plot of daily new cases against the internal mobility of each state for day 100. As one can see, there is some correlation between internal mobility and new cases. The correlation is particularly strong when the number of new cases is small, which is understandable as it fits the pattern of local community outbreaks of the epidemic. However, there are some strong outliers as well, especially when the numerical values of new cases is large, which also fits our expectation as we do not expect internal mobility to be the sole contributor to the spread of disease. As

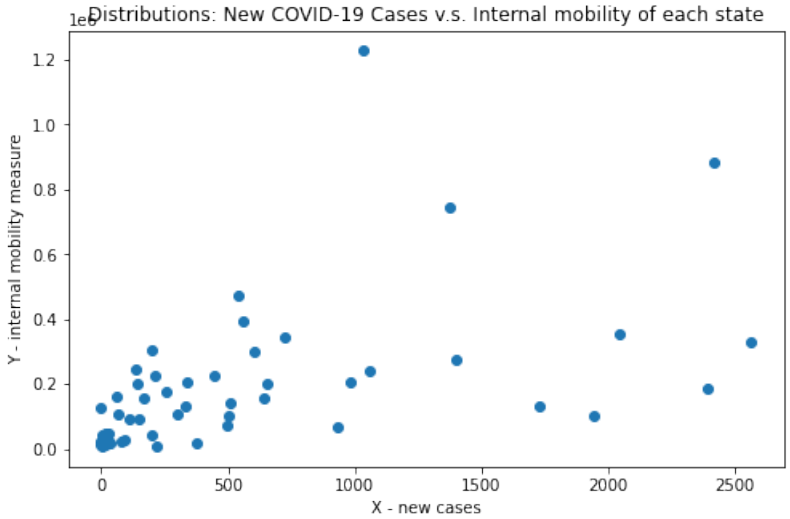


Figure 8: Daily new COVID-19 cases (delayed by 14 days) v.s. Internal mobility

shown in Figure.9, we have also checked the relationship between daily new cases of each state against external mobility of each state. The relationship is less clear compared to internal mobility. It is also worthy to note that *external mobility* is of orders of magnitude smaller than *internal mobility*. This is within our expectation, as the COVID-19 outbreak has limited inter-state travels significantly.

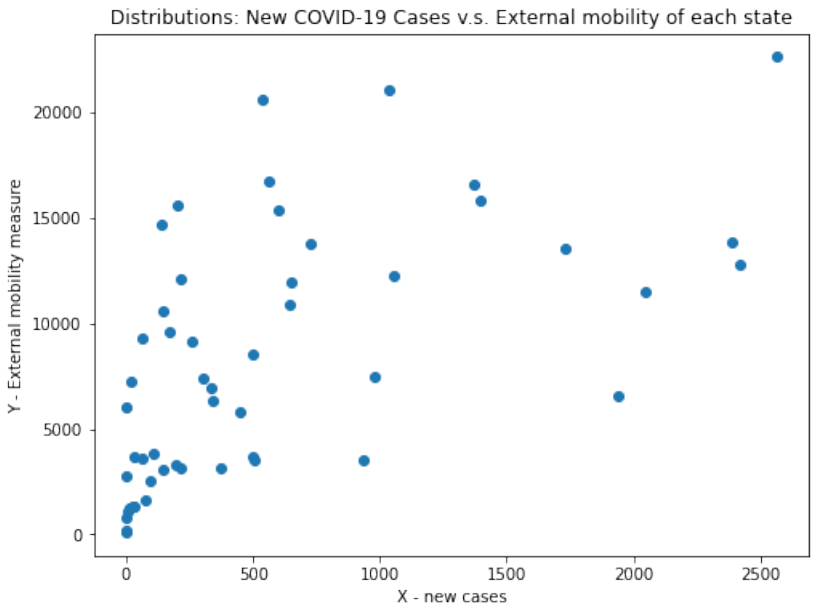


Figure 9: Daily new COVID-19 cases (delayed by 14 days) v.s. External mobility

As for the CDC dataset, we have noticed a large variance among states. In Figure. 10, we illustrate the average daily new cases of each U.S. State over the time period we intend to train our model for. The distribution is relatively sparse, with some significant outliers that have high COVID-19 cases on average consistently, while there is also a group of states that have low COVID-19 cases on average, probably because of the sparse distribution of population.

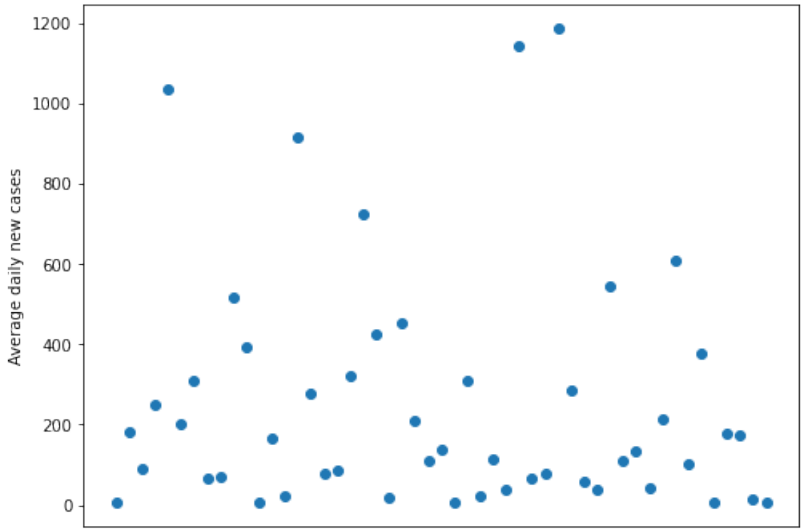


Figure 10: Average daily new cases of U.S. states

4.1.3 Retrospection

In retrospection, we have made a bold assumption where we believe aggregating the foot traffic of all POIs in that state can provide a good estimation for both *internal mobility* and *external mobility* of each state. This assumption seems to hold true

when it comes to *internal mobility*, while it is hard to determine if it is the case for *external mobility*. As the CDC pandemic tracker dataset only provides state-level data, we have chosen to pre-process everything at the state level. This leads to large *internal mobility* and small *external mobility*. The scale differences may impact our model in the training stage, which is an interesting follow-up project to investigate about.

4.2 Experiments

The data we use is from Jan/22/2020 (Denoted as Day 0) to June/15/2020. (Denoted as Day 145). The experiment procedure is, for each day T from Day 50 to Day 145, we use Day 0 Day T-10 for training, Day T-10 to Day T-9 for validation and Day T for testing. The test MAEs are averaged as the final test error. Note that, each time the model is initialized. Mean square error loss and Adam optimizer with $1e^{-3}$ learning rate are used for training. Models are supposed to run for 150 epochs and will stop early if there is no improvement for 30 epochs.

4.3 Results

The results are measured in terms of the mean absolute error (MAE) between predicted daily cases and true daily cases.

$$MAE = \frac{1}{N}\sum_{i}^{N} ||True| - |Pred||$$

The result is shown in Table. 3. The numbers are the MAE averaged across 51 regions. MPNN-LSTM-Attn has the best performance. MPNN-LSTM and MPNN-TSFM are slightly lower than that. It can also be observed in Fig. 11 that, the error of MPNN increases and the target date, while MPNN-LSTM and MPNN-LSTM-Attn are quite stable.

We also printed the attention weights of MPNN-LSTM-Attn, shown in Table. 6 and Fig. 12. The weights of seven days are almost equal. It means that there is no significant different among the features of different days. This result is expected because the time taken from being infected to having a positive test actually depends on multiple factors. Even though, we can still see a slight tendency in Fig. 12 that features of recent days have more influence on the final results.

	N+6	N+10	N+14	N+18	N+22
LSTM	356	356	355	354	357
MPNN	177	214	254	275	304
MPNN-LSTM	143	141	139	136	136
MPNN-LSTM-Attn	144	137	137	134	138
MPNN-TSFM	148	142	138	138	138

Table 3: Testing MAE on Day N+6, N+10, N+14, N+18 and N+22. MPNN-LSTM refers to MPNN-LSTM network, MPNN-LSTM-Attn refers to MPNN-LSTM with attention module, and MPNN-TSFM is MPNN with transorfmer.

	N-6	N-5	N-4	N-3	N-2	N-1	N
N+6	0.141	0.141	0.146	0.138	0.144	0.145	0.144
N+10	0.141	0.141	0.146	0.141	0.142	0.145	0.143
N+14	0.142	0.138	0.147	0.143	0.140	0.144	0.145
N+18	0.143	0.139	0.146	0.139	0.144	0.146	0.143
N+22	0.141	0.143	0.142	0.145	0.142	0.140	0.146

Table 4: Average attention weight of features from each day. For example, the first row contains each day's weight for model trained to predict Day N+6 case.

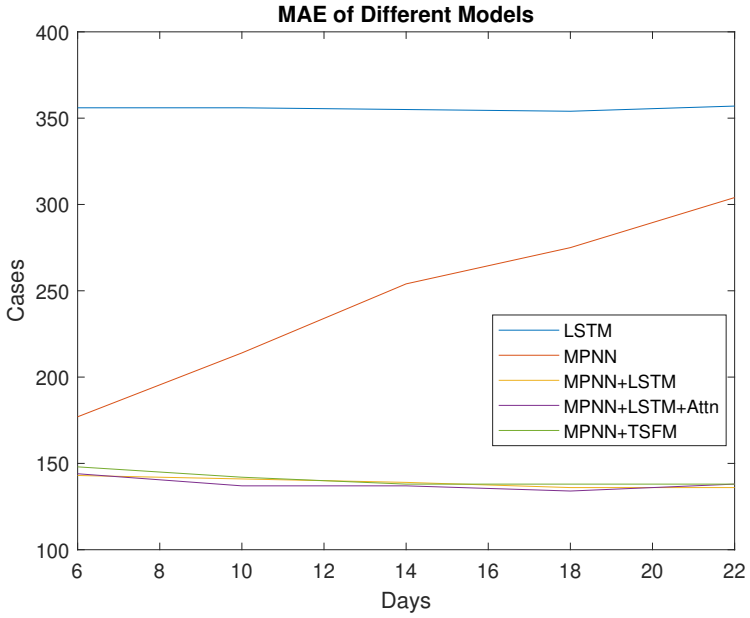


Figure 11: MAE of Different Models

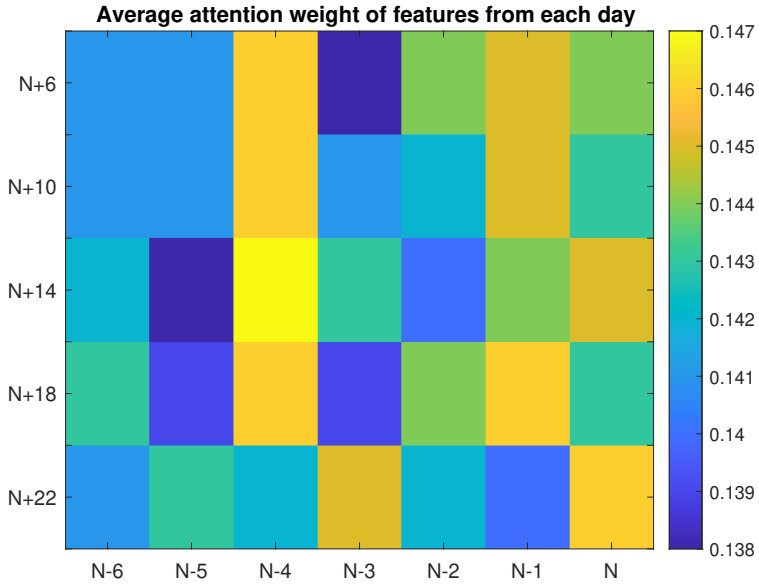
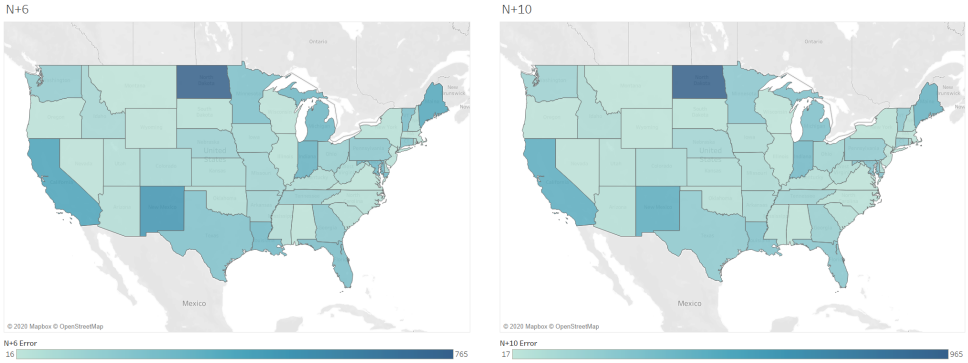
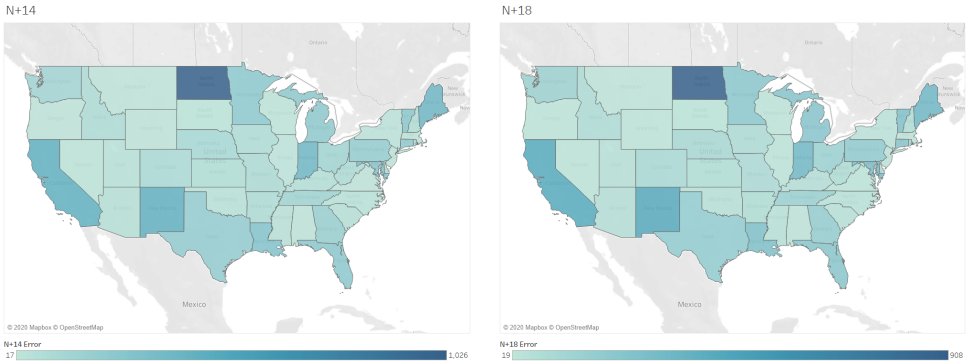


Figure 12: Average attention weight of features from each day



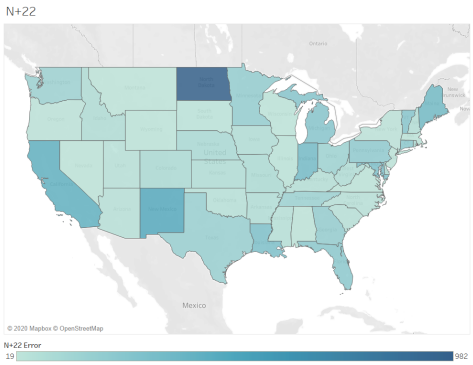
(a) N+6

(b) N+10



(c) N+14

(d) N+18



(e) N+22

Figure 13: Error of Different States using MPNN+LSTM model

5 Discussion and Future Work

In this paper, we experimented several models for COVID-19 forecasting utilizing mobility data and reported daily cases in the U.S. region in order to provide insights for public health decision makers in terms of appropriate interventions and resource allocation. The input data are derived from SafeGraph Mobility Data available until June 2020, and CDC Pandemic Trackers which provides new COVID-19 related cases and deaths on a daily basis on a state level. Proposed models include variations built upon recent works on Graph Neural Networks, where we built a graph with nodes as regions and edges defining mobility data between the endpoints. Models experimented include Message Passing Neural Networks (MPNN), Long-Short Term Memory network, MPNN-LSTM, MPNN-LSTM with a self-attention mechanism (Attn), and MPNN-Transformer (TSFM). The averaged mean absolute error (MAE) between predicted daily cases and true daily cases is used as the evaluation metric. Experiments conducted on 51 U.S. state-level regions demonstrate a relatively strong predictive power of MPNN-LSTM-Attn among all proposed models, where MPNN-LSTM and MPNN-TSFM achieve slightly higher MAE.

For future directions, we would like to explore in several aspects including data input, model building, and evaluation.

5.1 Data

In terms of the input data, it would be optimal if there exists data with more precise and updated mobility data. Since the mobility data we used from the SafeGraph only includes data until June 2020, we are not able to utilize the cases number after June, when the pandemic was spreading faster and the data should be more precise comparing to the data in the early stages. Also, we assume aggregating the foot traffic of all POIs in the state could represent a reasonable estimation for both internal and external mobility level. If we could find mobility data directly including interactions between states, the input data would definitely be more precise. Moreover, we would like to provide our model with more region-specific data, such as county-level mobility and new cases and deaths. Adding county-level data, we would be able to build graphs over graphs, where a state node is consisted of a graph of county nodes, enabling a more precise description of geological relationships.

5.2 Model

In terms of model building, we would like to experiment more settings of parameters and also more models. We have predicted the daily new cases of each state on the following 6th, 10th, 14th, 18th and 22th day. Since the MAE remains stable for different prediction time points for the MPNN-LSTM, MPNN-LSTM-Attn and MPNN-TSFM models, we could try a long prediction time period of 26th, 30th, 34th day, etc. However, we should also bear in mind that the effect of mobility on case numbers decreases as days increase. When trying to predict case numbers in a longer term, we would be interested in how the average attention weight of features would change in the MPNN-LSTM-Attn model, which is our best-performing model. Moreover, we could try adding the Model-Agnostic Meta-Learning (MAML) algorithm in addition to the MPNN model. In this case, we would be able to transform information obtained in a relatively stabilized region, for example, New York, to a region where the pandemic is still on the surging curve, for example, California.

5.3 Evaluation

In terms of the evaluation step, we would like to include more baseline and benchmark methods to compare our results with. Although MPNN alone and LSTM are already regarded as baselines for our variations on MPNN models, a set of more universal baselines or more recent works on pandemic prediction would provide more direct insights on whether our proposed models outperform other methods. Such baselines and benchmark method could possibly include average number of cases in the past days, a simple autoregressive moving average model with the input of the whole time-series of the region, the Prophet model, etc.

6 Work Distribution

Task	Members
Literature study	All members
Coding	Danfeng Guo, Yijing Zhou
Data preprocessing	Danfeng Guo, Yijing Zhou, Chenyang Wang, Haochen Yin
Model training	Haochen Yin, Chenyang Wang, Danfeng Guo, Yijing Zhou
Result Analysis	Haochen Yin, Chenyang Wang
Report Writing	All members

Table 5: Workload distribution among members

References

[1] S. Deng, S. Wang, H. Rangwala, L. Wang, and Y. Ning, "Graph message passing with cross-location attentions for long-term ili prediction," 2019.

[2] R. Trivedi, M. Farajtabar, P. Biswal, and H. Zha, "Dyrep: Learning representations over dynamic graphs," 2019.

[3] J. Qiu, J. Tang, H. Ma, Y. Dong, K. Wang, and J. Tang, "Deepinf: Social influence prediction with deep learning," 2018.

[4] J. Gao, R. Sharma, C. Qian, L. M. Glass, J. Spaeder, J. Romberg, J. Sun, and C. Xiao, "Stan: Spatio-temporal attention network for pandemic prediction using real world evidence," 2020.

[5] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," 2014.

[6] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," 2017.

[7] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2017.

[8] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *ICLR*, vol. abs/1409.0473, 2015.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.